

Oracle® Solaris 11.1 でのシステム情報、プロセス、およびパフォーマンスの管理

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

はじめに	7
1 システム情報の管理 (タスク)	11
システム情報の表示と変更に関する新機能	11
hostname および domainname コマンドの拡張機能	11
システム情報の表示	12
システム情報の表示 (タスクマップ)	12
システム情報の表示に使用されるコマンド	13
▼ システムのリリース情報を表示する方法	13
▼ 日付と時間を表示する方法	14
▼ システムのホスト ID を表示する方法	14
システムのアーキテクチャーおよびプロセッサタイプの表示	14
▼ システムの製品名を表示する方法	15
▼ システムにインストールされているメモリーを表示する方法	16
▼ デバイスのデフォルトのプロパティ値とカスタマイズされたプロパティ値を 表示する方法	16
▼ システムの診断情報を表示する方法	20
チップマルチスレッド化機能に関する情報の識別	22
▼ システムの物理プロセッサタイプを表示する方法	22
▼ システムの仮想プロセッサタイプを表示する方法	23
システム情報の変更	24
システム情報の変更 (タスクマップ)	25
▼ システムの日付と時間を手動で設定する方法	25
▼ その日のメッセージを設定する方法	26
▼ システムのアイデンティティの変更方法	26
2 システムプロセスの管理 (タスク)	27
システムプロセスの管理に関する新機能	27

擬似システムプロセス	27
システムプロセスの管理	28
システムプロセスの管理(タスクマップ)	28
システムプロセスを管理するコマンド	29
プロセスクラス情報の管理	38
プロセスクラス情報の管理(タスクマップ)	39
プロセスのスケジュール優先順位の変更(priocntl)	39
▼ プロセスクラスに関する基本情報を表示する方法(priocntl)	40
▼ プロセスのグローバル優先順位を表示する方法	40
▼ プロセスの優先順位を指定する方法(priocntl)	42
▼ タイムシェアリングプロセスのスケジューリングパラメータを変更する方法 (priocntl)	42
▼ プロセスのクラスを変更する方法(priocntl)	43
タイムシェアリングプロセスの優先順位の変更(nice)	44
▼ プロセスの優先順位を変更する方法(nice)	45
システムのプロセスに関するトラブルシューティング方法	46
3 システムパフォーマンスの監視(タスク)	47
システムパフォーマンスタスクに関する情報の参照箇所	47
システムパフォーマンスおよびシステムリソース	48
プロセスとシステムのパフォーマンス	48
システムパフォーマンスの監視	50
監視ツール	51
システムパフォーマンス情報の表示	51
システムパフォーマンス情報の表示(タスクマップ)	52
仮想メモリーの統計情報の表示(vmstat)	52
ディスク使用状況の表示(iostat)	56
ディスク容量統計情報の表示(df)	58
システム動作の監視	60
システム動作の監視(タスクマップ)	60
システム動作の監視(sar)	61
システム動作データの自動収集(sar)	78
4 システムタスクのスケジュール設定(タスク)	83
システムタスクを自動的に実行する方法	83

繰り返されるジョブのスケジューリング (crontab)	84
1つのジョブのスケジューリング (at)	84
システムタスクのスケジュール設定	85
crontab ファイルの作成と編集 (タスクマップ)	85
繰り返されるシステムタスクのスケジューリング (cron)	86
crontab ファイルの作成と編集	89
crontab ファイルの表示	90
crontab ファイルの削除	92
crontab コマンドの使用制御	93
at コマンドを使用したタスクのスケジューリング	96
at コマンドの使用 (タスクマップ)	96
1つのシステムタスクのスケジューリング (at)	97
5 システムコンソール、端末デバイス、および電源サービスの管理 (タスク)	103
システムコンソール、端末デバイス、および電源サービスの管理に関する新機能	103
システム電源サービスの管理方法の変更	103
システムコンソールとローカル接続された端末デバイスの管理	104
システムコンソールとローカル接続された端末デバイスを管理する SMF サービス	104
システム電源サービスの管理	107
システム電源の問題のトラブルシューティング	110
索引	111

はじめに

『Oracle Solaris 11.1 でのシステム情報、プロセス、およびパフォーマンスの管理』は、Oracle Solaris システム管理に関する重要な情報を提供するドキュメントセットの一部です。このガイドには、SPARC および x86 の両方のシステムに関する情報が含まれています。

本書は、読者が次のタスクを終了済みであることを前提としています。

- Oracle Solaris ソフトウェアのインストールが完了していること
- 使用する予定のすべてのネットワークソフトウェアを設定済み

システム管理者にとって重要と思われる Oracle Solaris の新機能については、各章の初めにある新機能に関するセクションを参照してください。

注 - この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャーを使用するシステムをサポートしています。サポートされるシステムは、Oracle Solaris OS: Hardware Compatibility Lists に記載されています。このドキュメントでは、プラットフォームにより実装が異なる場合は、それを特記します。

このドキュメントの x86 に関連する用語については、次を参照してください。

- x86 は、64 ビットおよび 32 ビットの x86 互換製品系列を指します。
- x64 は特に 64 ビット x86 互換 CPU を指します。
- 「32 ビット x86」は、x86 をベースとするシステムに関する 32 ビット特有の情報を指します。

サポートされるシステムについては、[Oracle Solaris OS: Hardware Compatibility Lists](#) を参照してください。

対象読者

このマニュアルは、Oracle Solaris 11 リリースを実行している 1 つまたは複数のシステムの管理を行うユーザーを対象にしています。本書を使用するには、UNIX のシステム管理について 1-2 年の経験が必要です。UNIX システム管理のトレーニングコースに参加することも役に立ちます。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P-1 表記上の規則

字体	説明	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	ブレースホルダ: 実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
<i>AaBbCc123</i>	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第 6 章を参照してください。 キャッシュは、ローカルに格納されるコピーです。 ファイルを保存しないでください。 注: いくつかの強調された項目は、オンラインでは太字で表示されます。

コマンド例のシェルプロンプト

Oracle Solaris OSに含まれるシェルで使用する、UNIXのデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solarisのリリースによって異なります。

表 P-2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	#
C シェル	machine_name%
C シェルのスーパーユーザー	machine_name#

◆◆◆ 第 1 章

システム情報の管理 (タスク)

この章では、一般的なシステム情報を表示および変更するために必要なタスクを示します。

この章では、柔軟な方法でシステムリソースの割り当て、監視、および制御を可能にするリソース管理については言及しません。リソース管理によるシステムリソースの管理については、『Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理』の第 1 章「リソース管理の紹介」を参照してください。

この章の内容は次のとおりです。

- 11 ページの「システム情報の表示と変更に関する新機能」
- 12 ページの「システム情報の表示」
- 24 ページの「システム情報の変更」

システム情報の表示と変更に関する新機能

hostname および domainname コマンドの拡張機能

このリリースでは、hostname および domainname コマンドが拡張され、hostname と domainname を永続的に設定することが簡単にできるようになりました。これらのコマンドを使用すると、対応する SMF プロパティおよび関連する SMF サービスも自動的に更新されます。

詳細は、26 ページの「システムのアイデンティティの変更方法」と hostname(1)、domainname(1M)、および nodename(4) のマニュアルページを参照してください。

システム情報の表示

次の表に、一般的なシステム情報を変更するコマンドを示します。

システム情報の表示(タスクマップ)

タスク	説明	参照先
システムのリリース情報を表示します。	Oracle Solaris リリースバージョンを特定するには、 <code>/etc/release</code> ファイルの内容を表示します。	13 ページの「システムのリリース情報を表示する方法」
システムの日付と時間を表示します。	<code>date</code> コマンドを使用し、システムの日付と時間を表示します。	14 ページの「日付と時間を表示する方法」
システムのホスト ID 番号を表示します。	<code>hostid</code> コマンドを使用し、システムのホスト ID を表示します。	14 ページの「システムのホスト ID を表示する方法」
システムのアーキテクチャーまたはプロセッサタイプを表示します。	システムのアーキテクチャータイプを表示するには、 <code>isainfo</code> コマンドを使用します。システムのプロセッサタイプを表示するには、 <code>isalist</code> コマンドを使用します。	14 ページの「システムのアーキテクチャーおよびプロセッサタイプの表示」
システムの製品名を表示します。	<code>prtconf -b</code> コマンドを使用し、システムの製品名を表示できます。	15 ページの「システムの製品名を表示する方法」
システムにインストールされているメモリーを表示します。	<code>prtconf</code> コマンドを使用し、システムにインストールされたメモリーに関する情報を表示します。	16 ページの「システムにインストールされているメモリーを表示する方法」
デバイスの元の値とデフォルト値を表示します。	<code>prtconf</code> コマンドと <code>-u</code> オプションを使用すると、デバイスのデフォルトのプロパティ値と更新されたプロパティ値の両方が表示されます。	16 ページの「デバイスのデフォルトのプロパティ値とカスタマイズされたプロパティ値を表示する方法」
システムの構成および診断情報を表示します。	システムの構成および診断情報を表示するには、 <code>prtdiag</code> コマンドおよび該当するオプションを使用します。	20 ページの「システムの診断情報を表示する方法」

タスク	説明	参照先
システムの物理および仮想プロセッサ情報を表示します。	<p><code>psrinfo -p</code> コマンドを使用し、システム上の物理プロセッサの合計数を表示します。</p> <p><code>psrinfo -pv</code> コマンドを使用すると、システム内の全物理プロセッサのほか、各物理プロセッサに関連した仮想プロセッサも表示されます。</p>	22 ページの「システムの物理プロセッサタイプを表示する方法」

システム情報の表示に使用されるコマンド

表 1-1 システム情報を表示するためのコマンド

コマンド	表示されるシステム情報	マニュアルページ
<code>date</code>	日付と時間	date(1)
<code>hostid</code>	ホスト ID	hostid(1)
<code>isainfo</code>	動作しているシステムのネイティブアプリケーションによってサポートされるビット数。ビット数は、トークンとしてスクリプトに渡すことができます。	isainfo(1)
<code>isalist</code>	プロセッサタイプ	isalist(1)
<code>prtconf</code>	システム構成情報、インストールされたメモリー、デバイスのプロパティ、および製品名	prtconf(1M)
<code>prtdiag</code>	障害の発生した現場交換可能ユニット (FRU) を含む、システムの構成および診断情報	prtdiag(1M)
<code>psrinfo</code>	プロセッサ情報	psrinfo(1M)
<code>uname</code>	オペレーティングシステム名、リリース、バージョン、ノード名、ハードウェア名、およびプロセッサタイプ	uname(1)

▼ システムのリリース情報を表示する方法

- リリースバージョンを特定するには、`/etc/release` ファイルの内容を表示します。

```
$ cat /etc/release
```

▼ 日付と時間を表示する方法

- システムクロックに従った現在の日付と時間を表示するには、**date** コマンドを使用します。

例 1-1 日付と時間を表示する

次の例は、date コマンドの出力を示します。

```
$ date
Fri Jun  1 16:07:44 MDT 2012
$
```

▼ システムのホスト ID を表示する方法

- ホスト ID を数値 (16 進) 形式で表示するには、**hostid** コマンドを使用します。

例 1-2 システムのホスト ID を表示する

次の例は、hostid コマンドの出力を示します。

```
$ hostid
80a5d34c
```

システムのアーキテクチャーおよびプロセッサタイプの表示

次の例は、x86 および SPARC ベースのシステムで実行された場合の、**isainfo** および **isalist** コマンドの出力を示しています。

例 1-3 システムのアーキテクチャータイプを表示する

次の例は、現在のオペレーティングシステムでサポートされているアプリケーションのネイティブ命令セットのアーキテクチャータイプと名前を表示する方法を示しています。

X86 ベースのシステムからの出力は次のとおりです。

```
$ isainfo
amd64 i386
```

SPARC ベースのシステムからの出力は次のとおりです。

```
$ isainfo
sparcv9 sparc
```

例1-3 システムのアーキテクチャタイプを表示する (続き)

`isainfo -v` コマンドは、32 ビットおよび64 ビットのアプリケーションサポートを表示します。たとえば、SPARC ベースのシステムからの出力は次のとおりです。

```
$ isainfo -v
64-bit sparcv9 applications
    asi_blk_init
32-bit sparc applications
    asi_blk_init v8plus div32 mul32
#
```

次の例は、x86 ベースのシステムからの `isainfo -v` コマンドの出力を示しています。

```
$ isainfo -v
64-bit amd64 applications
    sse4.1 ssse3 ahf cx16 sse3 sse2 sse fxsr mmx cmov amd_sysc cx8 tsc fpu
32-bit i386 applications
    sse4.1 ssse3 ahf cx16 sse3 sse2 sse fxsr mmx cmov sep cx8 tsc fpu
```

[isainfo\(1\)](#) のマニュアルページを参照してください。

例1-4 システムのプロセッサタイプを表示する

次の例は、x86 ベースのシステムのプロセッサタイプに関する情報を表示する方法を示しています。

```
$ isalist
pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

次の例は、SPARC ベースのシステムのプロセッサタイプに関する情報を表示する方法を示しています。

```
$ isalist
sparcv9 sparcv8plus sparcv8 sparcv8-fsmuld sparcv7 sparc sparcv9+vis sparcv9+vis2 \
sparcv8plus+vis sparcv8plus+vis2
```

[isalist\(1\)](#) のマニュアルページを参照してください。

▼ システムの製品名を表示する方法

`prtconf` コマンドの `-b` オプションを使用すると、システムの製品名を表示できます。詳細は、[prtconf\(1M\)](#) のマニュアルページを参照してください。

- システムの製品名を表示するには、`-b` オプションを付けて `prtconf` コマンドを使用します。

```
$ prtconf -b
```

例 1-5 SPARC: システムの製品名を表示する

次に、SPARC ベースのシステムでの `prtconf - b` コマンドの出力例を示します。

```
$ prtconf -b
name: ORCL,SPARC-T4-2
banner-name: SPARC T4-2
compatible: 'sun4v'
$
```

次に、SPARC ベースのシステムでの `prtconf - vb` コマンドの出力例を示します。

```
$ prtconf -vb
name: ORCL,SPARC-T3-4
banner-name: SPARC T3-4
compatible: 'sun4v'
idprom: 01840014.4fa02d28.00000000.a02d28de.00000000.00000000.00000000.00000000
openprom model: SUNW,4.33.0.b
openprom version: 'OBP 4.33.0.b 2011/05/16 16:26'
```

▼ システムにインストールされているメモリーを表示する方法

- システムにインストールされているメモリー容量を表示するには、`prtconf` コマンドを使用します。

例 1-6 システムにインストールされているメモリーを表示する

次の例は、`prtconf` コマンドの出力例を示します。`grep Memory` コマンドは `prtconf` コマンドの出力内容を選別して、メモリー情報だけを表示します。

```
$ prtconf | grep Memory
Memory size: 523776 Megabytes
```

▼ デバイスのデフォルトのプロパティー値とカスタマイズされたプロパティー値を表示する方法

デバイスのデフォルトのプロパティー値とカスタマイズされたプロパティー値の両方を表示するには、`prtconf` コマンドと `-u` オプションを使用します。このオプションの詳細については、[prtconf\(1M\)](#) のマニュアルページを参照してください。

- `driver.conf` ファイルのデフォルトのプロパティーとカスタマイズされたプロパティーを表示します。

```
$ prtconf -u
```

`prtconf -u` コマンドの出力には、システム上に存在するすべてのドライバのデフォルトのプロパティとカスタマイズされたプロパティが表示されます。

例 1-7 SPARC: デフォルトおよびカスタムのデバイスプロパティを表示する

この例は、`bge.conf` ファイルのデフォルトプロパティとカスタムプロパティを示しています。ベンダーから提供された構成ファイルは `/kernel` ディレクトリと `/platform` ディレクトリにありますが、対応する修正されたドライバ構成ファイルは `/etc/driver/drv` にあります。

```
$ prtconf -u
System Configuration: Oracle Corporation sun4v
Memory size: 523776 Megabytes
System Peripherals (Software Nodes):

ORCL,SPARC-T3-4
  scsi_vhci, instance #0
    disk, instance #4
    disk, instance #5
    disk, instance #6
    disk, instance #8
    disk, instance #9
    disk, instance #10
    disk, instance #11
    disk, instance #12
  packages (driver not attached)
    SUNW,builtin-drivers (driver not attached)
    deblocker (driver not attached)
    disk-label (driver not attached)
    terminal-emulator (driver not attached)
    dropins (driver not attached)
    SUNW,asr (driver not attached)
    kbd-translator (driver not attached)
    obp-tftp (driver not attached)
    zfs-file-system (driver not attached)
    hsfs-file-system (driver not attached)
  chosen (driver not attached)
  openprom (driver not attached)
    client-services (driver not attached)
  options, instance #0
  aliases (driver not attached)
  memory (driver not attached)
  virtual-memory (driver not attached)
  iscsi-hba (driver not attached)
    disk, instance #0 (driver not attached)
  virtual-devices, instance #0
    flashprom (driver not attached)
    tpm, instance #0 (driver not attached)
    n2cp, instance #0
    ncp, instance #0
    random-number-generator, instance #0
    console, instance #0
    channel-devices, instance #0
      virtual-channel, instance #0
      virtual-channel, instance #1
```

```

        virtual-channel-client, instance #2
        virtual-channel-client, instance #3
        virtual-domain-service, instance #0
cpu (driver not attached)

```

例 1-8 x86: デフォルトおよびカスタムのデバイスプロパティを表示する

この例は、`bge.conf` ファイルのデフォルトプロパティとカスタムプロパティを示しています。ベンダーから提供された構成ファイルは `/kernel` ディレクトリと `/platform` ディレクトリにありますが、対応する修正されたドライバ構成ファイルは `/etc/driver/drv` にあります。

```

$ prtconf -u
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
  scsi_vhci, instance #0
  pci_instance #0
    pci10de,5e (driver not attached)
  isa, instance #0
    asy, instance #0
    motherboard (driver not attached)
    pit_beep, instance #0
  pci10de,cb84 (driver not attached)
  pci108e,cb84, instance #0
    device, instance #0
      keyboard, instance #0
      mouse, instance #1
  pci108e,cb84, instance #0
  pci-ide, instance #0
    ide, instance #0
      sd, instance #0
    ide (driver not attached)
  pci10de,5c, instance #0
    display, instance #0
  pci10de,cb84, instance #0
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci1022,1100, instance #0
  pci1022,1101, instance #1
  pci1022,1102, instance #2
  pci1022,1103 (driver not attached)
  pci1022,1100, instance #3
  pci1022,1101, instance #4
  pci1022,1102, instance #5
  pci1022,1103 (driver not attached)

```

```

pci, instance #1
  pci10de,5e (driver not attached)
  pci10de,cb84 (driver not attached)
  pci10de,cb84, instance #1
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci1022,7458, instance #1
  pci1022,7459 (driver not attached)
  pci1022,7458, instance #2
    pci8086,1011, instance #0
    pci8086,1011, instance #1
    pci1000,3060, instance #0
      sd, instance #1
      sd, instance #2
    pci1022,7459 (driver not attached)
ioapics (driver not attached)
  ioapic, instance #0 (driver not attached)
  ioapic, instance #1 (driver not attached)
fw, instance #0
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  sb, instance #1
used-resources (driver not attached)
iscsi, instance #0
fcoe, instance #0
pseudo, instance #0
options, instance #0
xsvc, instance #0
vga_arbiter, instance #0

```

例 1-9 x86: システム構成情報の表示

次の例は、x86 ベースのシステムで `prtconf` コマンドを `-v` オプションとともに使用して、システムに接続されているディスク、テープ、および DVD デバイスを識別する方法を示しています。このコマンドの出力では、デバイスインスタンスの出力の横に `driver not attached` メッセージが表示されます。このメッセージは通常、そのデバイスインスタンスにデバイスがないことを示します。

```

$ prtconf -v | more
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
  System properties:
    name='#size-cells' type=int items=1
      value=00000002
    name='#address-cells' type=int items=1
      value=00000003
    name='relative-addressing' type=int items=1
      value=00000001
    name='MMU_PAGEOFFSET' type=int items=1

```

```

        value=00000fff
name='MMU_PAGESIZE' type=int items=1
        value=00001000
name='PAGESIZE' type=int items=1
        value=00001000
name='acpi-status' type=int items=1
        value=00000013
name='biosdev-0x81' type=byte items=588
        value=01.38.74.0e.08.1e.db.e4.fe.00.d0.ed.fe.f8.6b.04.08.d3.db.e4.fe
.
.
.

```

参照 詳細は、[driver\(4\)](#)、[driver.conf\(4\)](#)、および [prtconf\(1M\)](#) のマニュアルページを参照してください。

管理者が提供する構成ファイルの作成手順については、『[Oracle Solaris 11.1 の管理: デバイスとファイルシステム](#)』の第3章「[デバイスの管理 \(タスク\)](#)」を参照してください。

▼ システムの診断情報を表示する方法

- システムの構成および診断情報を表示します。

```
$ prtdiag [-v] [-l]
```

-v 冗長モード。

-l ログ出力。システムに障害やエラーがある場合は、この情報を [syslogd\(1M\)](#) のみに出力します。

例 1-10 SPARC: システムの診断情報を表示する

次の例は、SPARC ベースのシステムでの `prtdiag -v` コマンドの出力を示しています。簡潔にするために、この例は省略されています。

```
$ prtdiag -v | more
```

```
System Configuration: Oracle Corporation sun4v Sun Fire T200
Memory size: 16256 Megabytes
```

```
===== Virtual CPUs =====
```

CPU ID	Frequency	Implementation	Status
0	1200 MHz	SUNW,UltraSPARC-T1	on-line
1	1200 MHz	SUNW,UltraSPARC-T1	on-line
2	1200 MHz	SUNW,UltraSPARC-T1	on-line
3	1200 MHz	SUNW,UltraSPARC-T1	on-line
4	1200 MHz	SUNW,UltraSPARC-T1	on-line
5	1200 MHz	SUNW,UltraSPARC-T1	on-line


```

DDR2      in use 0   DIMM3      NODE0
unknown   empty 0   DIMM0      NODE1
unknown   empty 0   DIMM1      NODE1
DDR2      in use 0   DIMM2      NODE1
DDR2      in use 0   DIMM3      NODE1

==== On-Board Devices =====
LSI serial-SCSI #1
Gigabit Ethernet #1
ATI Rage XL VGA

==== Upgradeable Slots =====

ID  Status   Type           Description
---  -
1   available PCI Express   PCIExp SLOT0
2   available PCI Express   PCIExp SLOT1
3   available PCI-X         PCIX SLOT2
4   available PCI Express   PCIExp SLOT3
5   available PCI Express   PCIExp SLOT4
$

```

チップマルチスレッド化機能に関する情報の識別

`psrinfo` コマンドが変更され、仮想プロセッサの情報だけでなく、物理プロセッサの情報も返すようになりました。この拡張機能の追加によって、チップマルチスレッド化 (CMT) 機能を識別できるようになりました。新しい `-p` オプションは、システム内の物理プロセッサの合計数を報告します。`psrinfo -pv` コマンドを使用すると、システム内の全物理プロセッサのほか、各物理プロセッサに関連した仮想プロセッサも表示されます。`psrinfo` コマンドのデフォルト出力は、これまでどおりシステムの仮想プロセッサ情報を表示します。

詳細は、[psrinfo\(1M\)](#) のマニュアルページを参照してください。

この機能に関連する手順については、[22 ページ](#)の「システムの物理プロセッサタイプを表示する方法」を参照してください。

▼ システムの物理プロセッサタイプを表示する方法

- `psrinfo -p` コマンドを使用し、システム上の物理プロセッサの合計数を表示します。

```

$ psrinfo -p
1

```

psrinfo -pv コマンドを使用すると、システム上の各物理プロセッサのほか、各物理プロセッサに関連した仮想プロセッサの情報も表示されます。例:

```
$ psrinfo -pv
The physical processor has 8 cores and 32 virtual processors (0-31)
  The core has 4 virtual processors (0-3)
    The core has 4 virtual processors (4-7)
    The core has 4 virtual processors (8-11)
    The core has 4 virtual processors (12-15)
    The core has 4 virtual processors (16-19)
    The core has 4 virtual processors (20-23)
    The core has 4 virtual processors (24-27)
    The core has 4 virtual processors (28-31)
      UltraSPARC-T1 (chipid 0, clock 1000 MHz)
```

psrinfo -pv コマンドを x86 ベースのシステムで使用すると、以下の出力が表示されます。

```
$ psrinfo -pv
The physical processor has 2 virtual processors (0 1)
  x86 (AuthenticAMD 40F13 family 15 model 65 step 3 clock 2793 MHz)
    Dual-Core AMD Opteron(tm) Processor 2220      [ Socket: F(1207) ]
The physical processor has 2 virtual processors (2 3)
  x86 (AuthenticAMD 40F13 family 15 model 65 step 3 clock 2793 MHz)
    Dual-Core AMD Opteron(tm) Processor 2220      [ Socket: F(1207) ]
```

▼ システムの仮想プロセッサタイプを表示する方法

- システムの仮想プロセッサタイプに関する情報を表示するには、**psrinfo -v** コマンドを使用します。

```
$ psrinfo -v
```

x86 ベースのシステムでは、isalist コマンドを使用し、仮想プロセッサタイプを表示します。例:

```
$ isalist
amd64 pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

例 1-12 SPARC: システムの仮想プロセッサタイプを表示する

この例は、SPARC ベースのシステムの仮想プロセッサタイプに関する情報を表示する方法を示しています。

```
$ psrinfo -v
Status of virtual processor 28 as of: 09/13/2010 14:07:47
  on-line since 04/08/2010 21:27:56.
    The sparcv9 processor operates at 1400 MHz,
      and has a sparcv9 floating point processor.
Status of virtual processor 29 as of: 09/13/2010 14:07:47
```

```
on-line since 04/08/2010 21:27:56.  
The sparcv9 processor operates at 1400 MHz,  
and has a sparcv9 floating point processor.
```

例 1-13 SPARC: システムの各物理プロセッサに関連付けられている仮想プロセッサを表示する

次の例は、Oracle SPARC T4-4 サーバーで `-pv` オプションとともに実行された場合の、`psrinfo` コマンドの出力を示しています。出力には、チップ(物理プロセッサ)と、スレッド位置に関するコア情報の両方が表示されます。この情報は、スレッドがどの物理CPU 上にあるか、およびコアレベルでどのようにマップされているかを調べるのに役立ちます。

```
$ psrinfo -pv  
The physical processor has 8 cores and 64 virtual processors (0-63)  
  The core has 8 virtual processors (0-7)  
  The core has 8 virtual processors (8-15)  
  The core has 8 virtual processors (16-23)  
  The core has 8 virtual processors (24-31)  
  The core has 8 virtual processors (32-39)  
  The core has 8 virtual processors (40-47)  
  The core has 8 virtual processors (48-55)  
  The core has 8 virtual processors (56-63)  
  SPARC-T4 (chipid 0, clock 2998 MHz)  
The physical processor has 8 cores and 64 virtual processors (64-127)  
  The core has 8 virtual processors (64-71)  
  The core has 8 virtual processors (72-79)  
  The core has 8 virtual processors (80-87)  
  The core has 8 virtual processors (88-95)  
  The core has 8 virtual processors (96-103)  
  The core has 8 virtual processors (104-111)  
  The core has 8 virtual processors (112-119)  
  The core has 8 virtual processors (120-127)  
  SPARC-T4 (chipid 1, clock 2998 MHz)
```

システム情報の変更

このセクションでは、一般的なシステム情報を変更するコマンドを説明します。

システム情報の変更(タスクマップ)

タスク	説明	参照先
システムの日付と時間を手動で設定します。	システムの日付と時間を手動で設定するには、 <code>date mmdhHHMM[[cc]yy]</code> コマンド行構文を使用します。	25 ページの「システムの日付と時間を手動で設定する方法」
その日のメッセージを設定します。	システムのその日のメッセージを設定するには、 <code>/etc/motd</code> ファイルを編集します。	26 ページの「その日のメッセージを設定する方法」
システムのアイデンティティを変更します。	<code>hostname</code> コマンドを使用して、システムのアイデンティティを変更します。	26 ページの「システムのアイデンティティの変更方法」

▼ システムの日付と時間を手動で設定する方法

- 1 管理者になります。
『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。
- 2 次のように新しい日付と時間を入力します。

```
$ date mmdhHHMM[[cc]yy]
```

mm 月。2桁を使用します。
dd 日。2桁を使用します。
HH 時。2桁で24時間制を使用します。
MM 分。2桁を使用します。
cc 世紀。2桁を使用します。
yy 年。2桁を使用します。

詳細については、[date\(1\)](#)のマニュアルページを参照してください。
- 3 オプションを付けずに `date` コマンドを使用し、システム日付が正しく設定されたことを確認します。

例 1-14 システムの日付と時間を手動で設定する

次の例は、`date` コマンドを使用して手作業でシステムの日付と時間を設定する方法を示します。

```
# date
Monday, September 13, 2010 02:00:16 PM MDT
# date 0921173404
Thu Sep 17:34:34 MST 2010
```

▼ その日のメッセージを設定する方法

その日のメッセージファイル `/etc/motd` を編集して、システム的全ユーザーに対して、ログイン時に通知または問い合わせる内容を書き込むことができます。ただし、この機能を使用するときは、必要なメッセージだけを送るようにします。メッセージファイルは定期的に編集して、不用になったメッセージを削除することをお勧めします。

- 1 **root** 役割になるか、**Administrator Message Edit** プロファイルが割り当てられている役割になります。

『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 **pfedit** コマンドを使用して `/etc/motd` ファイルを編集し、必要なメッセージを追加します。

```
$ pfedit /etc/motd
```

テキストを編集して、ユーザーのログイン時に表示されるメッセージを記述します。スペース、タブ、リターンも含めます。

- 3 `/etc/motd` ファイルの内容を表示して、変更内容を確認します。

```
$ cat /etc/motd
Welcome to the UNIX Universe. Have a nice day.
```

▼ システムのアイデンティティーの変更方法

- 1 **root** 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 システムのホスト名を設定します。

```
# hostname mynodename
```

システムプロセスの管理 (タスク)

この章では、システムプロセスを管理する手順について説明します。

この章の内容は次のとおりです。

- 27 ページの「システムプロセスの管理に関する新機能」
- 28 ページの「システムプロセスの管理」
- 38 ページの「プロセスクラス情報の管理」
- 46 ページの「システムのプロセスに関するトラブルシューティング方法」

システムプロセスの管理に関する新機能

このリリースでは、システムプロセスの管理に関する次の機能が新しく導入または変更されています。

擬似システムプロセス

Oracle Solaris 10 および Oracle Solaris 11 リリースには、特定のタスクを実行するが管理を必要としないシステムプロセスが含まれています。

プロセス	説明
fsflush	ページをディスクにフラッシュするシステムデーモン
init	その他のプロセスおよび SMF コンポーネントを起動および再起動する初期システムプロセス
intrad	割り込みによるシステム負荷を監視および分散するシステムプロセス
kmem_task	メモリーキャッシュのサイズを監視するシステムプロセス

プロセス	説明
pageout	ディスクへのメモリーページングを制御するシステムプロセス
sched	OSスケジューリングとプロセススワップを担当するシステムプロセス
vm_tasks	パフォーマンス向上のために仮想メモリー関連のワークロードを複数のCPUに分散する、プロセッサごとに1つのスレッドを持つシステムプロセス。
zpool-pool-name	関連するプールに対応するI/O taskq スレッドを含むZFSストレージプールごとのシステムプロセス

システムプロセスの管理

このセクションでは、システムプロセスを管理するためのさまざまなタスクについて説明します。

システムプロセスの管理 (タスクマップ)

タスク	説明	参照先
プロセスを表示する	ps コマンドを使用して、システム上のすべてのプロセスを表示する	32 ページの「プロセスを表示する方法」
プロセスについての情報を表示する	pgrep コマンドを使用して、詳細情報を表示したいプロセスのプロセスIDを取得する	33 ページの「プロセスに関する情報を表示する方法」
プロセスを制御する。	pgrep コマンドを使用して、プロセスを見つける。その後、適切な pcommand (/proc) を使用し、プロセスを制御する。(表 2-3) コマンドについては、Table 2-3 を参照。	35 ページの「プロセスを制御する方法」
プロセスを強制終了する。	プロセス名かプロセスIDを使用し、プロセスを見つける。pkill コマンドまたはkill コマンドを使用し、プロセスを終了する	35 ページの「プロセスを終了させる方法 (pkill)」 36 ページの「プロセスを終了させる方法 (kill)」

システムプロセスを管理するコマンド

次の表では、プロセス情報を管理するために使用されるコマンドについて説明します。

表 2-1 プロセスを管理するためのコマンド

コマンド	説明	マニュアルページ
ps, pgrep, prstat, pkill	システム上のアクティブなプロセスのステータスをチェックします。また、プロセスについての詳細な情報を表示します。	ps(1)、pgrep(1)、および prstat(1M)
pkill	pgrep と同様に機能するが、名前または他の属性によってプロセスを検索またはシグナルを送信して、プロセスを終了します。一致したプロセスのプロセス ID を出力するのではなく、kill コマンドと同様に、各プロセスにシグナルを送信します。	pgrep(1) および pkill(1) kill(1)
pargs, preap	プロセスのデバッグを支援します。	pargs(1) および preap(1)
dispadmin	デフォルトのプロセススケジューリングポリシーを表示します。	dispadmin(1M)
priocntl	プロセスに優先順位クラスを割り当てて、プロセスの優先度を管理します。	priocntl(1)
nice	タイムシェアリングプロセスの優先度を変更します。	nice(1)
psrset	特定のプロセスグループを、1つのプロセッサではなく、プロセッサのグループに結合します。	psrset(1M)

ps コマンドの使用

ps コマンドを使用すると、システム上で活動中のプロセスのステータスをチェックできます。また、プロセスについての技術的な情報も表示できます。このデータは、プロセスの優先順位の設定方法を決定するときなど、各種の管理タスクに利用できます。

使用するオプションに応じて、ps コマンドは以下の情報を報告します。

- プロセスの現在のステータス
- プロセス ID
- 親プロセス ID
- ユーザー ID
- スケジューリングクラス
- 優先順位
- プロセスのアドレス
- 使用したメモリー
- 使用した CPU 時間

次の表では、ps コマンドを使用して報告されるいくつかのフィールドについて説明します。どのフィールドが表示されるかは、選択するオプションによって異なります。使用可能なすべてのオプションについては、ps(1) のマニュアルページを参照してください。

表 2-2 ps により出力されるフィールドのサマリー

フィールド	説明
UID	プロセス所有者の実効ユーザー ID
PID	プロセス ID
PPID	親プロセス ID
C	スケジューリングのためのプロセッサ使用率。このフィールドは -c オプションを使用すると表示されない
CLS	プロセスが所属するスケジューリングクラス。リアルタイム、システム、またはタイムシェアリングのいずれか。このフィールドは、-c オプションを指定した場合にのみ表示される
PRI	カーネルスレッドのスケジューリング優先順位。番号が大きいほど優先順位が高い
NI	プロセスの nice 値。これは、スケジューリング優先順位に影響する。プロセスの nice 値を大きくすると、その優先順位が下がる
ADDR	proc 構造体のアドレス
SZ	プロセスの仮想アドレスサイズ
WCHAN	プロセスが休眠中のイベントまたはロックのアドレス
STIME	プロセスの起動時間 (時、分、秒)
TTY	プロセス (またはその親プロセス) が起動された端末。疑問符は、制御端末がないことを示す
TIME	プロセスの起動以降に使用した合計 CPU 時間

表 2-2 ps により出力されるフィールドのサマリー (続き)

フィールド	説明
CMD	プロセスを生成したコマンド

/proc ファイルシステムとコマンド

プロセスコマンドを使用すると、/proc ディレクトリにあるプロセスに関する詳細情報を表示できます。次の表に、/proc プロセスコマンドを示します。/proc ディレクトリはプロセスファイルシステム (PROCFS) とも呼ばれます。アクティブなプロセスのイメージは、そのプロセス ID 番号を使って /proc に格納されます。

表 2-3 プロセスコマンド (/proc)

プロセスコマンド	説明
pcred	プロセスの資格情報を表示する
pfiles	プロセス内で開いているファイルに関する fstat 情報と fcntl 情報を表示する
pflags	/proc 追跡フラグ、保留状態のシグナルと保持状態のシグナル、他のステータス情報を出力する
pldd	プロセスにリンクされている動的ライブラリを表示する
pmap	各プロセスのアドレス空間マップを表示する
psig	各プロセスのシグナルの動作とハンドラを表示する
prun	各プロセスを開始する
pstack	各プロセス内の LWP の 16 進とシンボルのスタックトレースを出力する
pstop	各プロセスを停止する
ptime	microstate アカウントを使用してプロセスの時間を測定する
ptree	プロセスを含むプロセスツリーを表示する
pwait	プロセス終了後のステータス情報を表示する
pwdx	プロセスの現在の作業ディレクトリを表示する

詳細については、[proc\(1\)](#) を参照してください。

プロセスツールは ps コマンドの一部のオプションに似ていますが、このツールから提供される出力の方が詳細です。

プロセスコマンドには次の機能があります。

- `fstat` や `fcntl`、作業ディレクトリ、親プロセスと子プロセスからなるツリーなど、プロセスに関する詳細情報を表示します。
- ユーザーがプロセスを停止または再開できるように、プロセスに対する制御を提供します。

プロセスコマンド (`/proc`) を使用したプロセスの管理

一部のプロセスコマンドを使用して、プロセスに関する詳細な技術情報を表示したり、アクティブなプロセスを制御したりできます。表 2-3 に一部の `/proc` コマンドを示しています。

プロセスが無限ループに陥った場合や、実行時間が長すぎる場合は、プロセスを終了 (`kill`) できます。 `kill` コマンドまたは `pkill` コマンドを使用してプロセスを終了する方法については、第 2 章「システムプロセスの管理 (タスク)」を参照してください。

`/proc` ファイルシステムは、状態情報と制御機能のためのサブディレクトリを含むディレクトリ階層です。

`/proc` ファイルシステムは、`xwatchpoint` 機能も提供します。この機能は、プロセスのアドレス空間の個々のページの読み取り権または書き込み権を再マップするために使用されます。この機能は制限がなく、MT-safe です。

デバッグ用ツールは、`/proc` の `xwatchpoint` 機能を使用するように変更されています。つまり、`xwatchpoint` プロセス全体がより高速になっています。

`dbx` デバッグ用ツールを使用して `xwatchpoint` を設定する際の、次の制限はなくなりました。

- SPARC レジスタウィンドウのため、スタック上のローカル変数に `xwatchpoint` を設定する。
- マルチスレッド化されたプロセスに `xwatchpoint` を設定する。

詳細については、`proc(4)` および `mdb(1)` のマニュアルページを参照してください。

▼ プロセスを表示する方法

- `ps` コマンドを使用すると、システム上の全プロセスを表示できます。

```
$ ps [-efc]
```

```
ps      ログインセッションに関連するプロセスのみを表示する
```

```
-ef     システム上で実行中のすべてのプロセスに関する詳細情報を表示する
```

```
-c      プロセススケジューラ情報を表示する
```

例 2-1 プロセスを表示する

次の例は、オプションを指定しないときの `ps` コマンドからの出力を示します。

```
$ ps
  PID TTY          TIME CMD
 1664 pts/4        0:06 csh
 2081 pts/4        0:00 ps
```

次の例は、`ps -ef` コマンドの出力を示します。この出力例は、システムのブート時に最初に実行されたプロセスが `sched` (スワップ) であり、それに続いて `init` プロセス、`pageout` の順に実行されたことを示しています。

```
$ ps -ef
UID    PID  PPID  C    STIME TTY          TIME CMD
root    0    0    0  18:04:04 ?        0:15 sched
root    5    0    0  18:04:03 ?        0:05 zpool-rpool
root    1    0    0  18:04:05 ?        0:00 /sbin/init
root    2    0    0  18:04:05 ?        0:00 pageout
root    3    0    0  18:04:05 ?        2:52 fsflush
root    6    0    0  18:04:05 ?        0:02 vmtasks
daemon 739   1    0  19:03:58 ?        0:00 /usr/lib/nfs/nfs4cbd
root    9    1    0  18:04:06 ?        0:14 /lib/svc/bin/svc.startd
root   11    1    0  18:04:06 ?        0:45 /lib/svc/bin/svc.configd
daemon 559   1    0  18:04:49 ?        0:00 /usr/sbin/rpcbind
netcfg  47    1    0  18:04:19 ?        0:01 /lib/inet/netcfgd
dladm  44    1    0  18:04:17 ?        0:00 /sbin/dlmgmt
netadm  51    1    0  18:04:22 ?        0:01 /lib/inet/ipmgmt
root   372  338   0  18:04:43 ?        0:00 /usr/lib/hal/hald-addon-cpufreq
root   67    1    0  18:04:30 ?        0:02 /lib/inet/in.mpathd
root  141    1    0  18:04:38 ?        0:00 /usr/lib/pfexecd
netadm  89    1    0  18:04:31 ?        0:03 /lib/inet/nwamd
root   602   1    0  18:04:50 ?        0:02 /usr/lib/inet/inetd start
root  131    1    0  18:04:35 ?        0:01 /sbin/dhcpagent
daemon 119    1    0  18:04:33 ?        0:00 /lib/crypto/kcfd
root   333   1    0  18:04:41 ?        0:07 /usr/lib/hal/hald --daemon=yes
root   370  338   0  18:04:43 ?        0:00 /usr/lib/hal/hald-addon-network-discovery
root   159    1    0  18:04:39 ?        0:00 /usr/lib/sysevent/syseventd
root   236    1    0  18:04:40 ?        0:00 /usr/lib/ldoms/drd
root   535    1    0  18:04:46 ?        0:09 /usr/sbin/nscd
root   305    1    0  18:04:40 ?        0:00 /usr/lib/zones/zonestatd
root   326    1    0  18:04:41 ?        0:03 /usr/lib/devfsadm/devfsadmd
root   314    1    0  18:04:40 ?        0:00 /usr/lib/dbus-daemon --system
```

▼ プロセスに関する情報を表示する方法

- 1 詳細を表示したいプロセスのプロセス ID を表示します。

```
# pgrep process
```

`process` は、詳細を表示したいプロセスの名前です。

プロセスIDは、出力の第1列目に表示されます。

- 表示したいプロセス情報を表示します。

```
# /usr/bin/pcommand pid
```

pcommand 実行したい(/proc)コマンド。表2-3にこれらのコマンドを示して説明しています。

pid プロセスID

例2-2 プロセスに関する情報を表示する

次の例は、プロセスコマンドを使用して cron プロセスに関する詳細情報を表示する方法を示しています。

```
# pgrep cron      1
4780
# pwdx 4780      2
4780: /var/spool/cron/atjobs
# ptree 4780     3
4780 /usr/sbin/cron
# pfiles 4780   4
4780: /usr/sbin/cron
Current rlimit: 256 file descriptors
0: S_IFCHR mode:0666 dev:290,0 ino:6815752 uid:0 gid:3 rdev:13,2
  O_RDONLY|O_LARGEFILE
  /devices/pseudo/mm@0:null
1: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
  O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
  /var/cron/log
2: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
  O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
  /var/cron/log
3: S_IFIFO mode:0600 dev:32,128 ino:42049 uid:0 gid:0 size:0
  O_RDWR|O_LARGEFILE
  /etc/cron.d/FIFO
4: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
  O_RDWR|O_NONBLOCK
5: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
  O_RDWR
```

- cron プロセスのプロセスIDを表示する。
- cron プロセスの現在の作業ディレクトリを表示する
- cron プロセスを含むプロセスツリーを表示する
- fstat と fcntl の情報を表示する

▼ プロセスを制御する方法

- 1 制御するプロセスのプロセスIDを表示します。

```
# pgrep process
```

process は、制御するプロセスの名前です。

プロセスIDは、出力の第1列目に表示されます。

- 2 適切なプロセスコマンドを使用してプロセスを制御します。

```
# /usr/bin/pcommand pid
```

pcommand 実行したいプロセス (/proc) コマンド。表 2-3 にこれらのコマンドを示して説明しています。

pid プロセスID

- 3 プロセスステータスを確認します。

```
# ps -ef | grep pid
```

プロセスの終了 (pkill, kill)

プロセスを強制的に終了 (kill) しなければならない場合があります。プロセスが無限ループに入る場合があります。大きいジョブを開始したが完了する前に停止したい場合があります。所有しているプロセスであれば、どれでも終了できます。また、スーパーユーザーはプロセスIDが0、1、2、3、および4のものを除き、システム上のどんなプロセスでも終了できます。プロセスIDが0、1、2、3、4のプロセスを終了させると、システムがクラッシュする可能性があります。

詳細については、[pgrep\(1\)](#)、[pkill\(1\)](#)、および[kill\(1\)](#)のマニュアルページを参照してください。

▼ プロセスを終了させる方法 (pkill)

- 1 ほかのユーザーのプロセスを終了するには、**root** 役割になります。

- 2 終了したいプロセスのプロセスIDを表示します。

```
$ pgrep process
```

process は、終了したいプロセスの名前です。

例:

```
$ pgrep netscape
587
566
```

プロセスIDが出力に表示されます。

注 - Sun Ray に関するプロセス情報を取得するには、次のコマンドを使用します。

```
# ps -fu user
```

このコマンドは、このユーザーのすべてのプロセスを表示します。

```
# ps -fu user | grep process
```

このコマンドは、ユーザーの特定のプロセスを検索します。

3 プロセスを終了します。

```
$ pkill [signal] process
```

signal pkill コマンド行構文にシグナルが何も含まれない場合は、使用されるデフォルトシグナルは -15 (SIGKILL)。-9 シグナル (SIGTERM) を pkill コマンドで使用すると、プロセスをただちに終了できる。ただし -9 シグナルは、データベースプロセスや LDAP サーバープロセスなどのプロセスを終了するために使用してはならない。データが失われる可能性がある

process 停止するプロセスの名前

ヒント - pkill コマンドを使用してプロセスを終了する場合は、まず、シグナルオプションを使用せずにコマンドだけで試行してみます。数分待つてプロセスが終了しなければ、pkill コマンドに -9 シグナルを付けて使用します。

4 プロセスが終了したことを確認します。

```
$ pgrep process
```

終了したプロセスは、pgrep コマンドの出力に表示されないはずです。

▼ プロセスを終了させる方法 (kill)

1 ほかのユーザーのプロセスを終了するには、root 役割になります。

2 終了したいプロセスのプロセスIDを表示します。

```
# ps -fu user
```

user は、プロセスを表示したいユーザーです。

プロセスIDは、出力の第1列目に表示されます。

3 プロセスを終了します。

```
# kill [signal-number] pid
```

signal pkill コマンド行構文にシグナルが何も含まれない場合は、使用されるデフォルトシグナルは -15 (SIGKILL)。-9 シグナル (SIGTERM) を pkill コマンドで使用すると、プロセスをただちに終了できる。ただし -9 シグナルは、データベースプロセスや LDAP サーバプロセスなどのプロセスを終了するために使用してはならない。データが失われる可能性がある

pid 終了したいプロセスのプロセス ID

ヒント - kill コマンドを使用してプロセスを終了する場合は、まず、シグナルオプションを使用せずにコマンドだけで試行してみます。数分待ってプロセスが終了しなければ、kill コマンドに -9 シグナルを付けて使用します。

4 プロセスが終了したことを確認します。

```
$ pgrep pid
```

終了したプロセスは、pgrep コマンドの出力に表示されないはずです。

プロセスのデバッグ (pargs, preap)

pargs コマンドと preap コマンドは、プロセスのデバッグを改善します。pargs コマンドを使用すると、動作中のプロセスまたはコアファイルに関連付けられた引数と環境変数を表示できます。preap コマンドを使用すると、終了した(ゾンビ)プロセスを削除できます。ゾンビプロセスとは、その終了ステータスがまだ親に回収されていないプロセスをいいます。これらのプロセスは概して無害ですが、数が多ければシステムリソースを消費します。pargs コマンドと preap コマンドを使用して、ユーザーの検査権限に含まれるすべてのプロセスを検査できます。スーパーユーザーは、すべてのプロセスを検査できます。

preap コマンドの使用方法については、[preap\(1\)](#) のマニュアルページを参照してください。pargs コマンドの使用方法については、[pargs\(1\)](#) のマニュアルページを参照してください。また、[proc\(1\)](#) のマニュアルページも参照してください。

例 2-3 プロセスをデバッグする (pargs)

pargs コマンドによって、プロセスに渡された引数を ps コマンドで一部しか表示できないという問題がようやく解決されました。次の例は、pargs コマンドを pgrep コマンドと併用して、プロセスに渡された引数を表示する方法を示します。

```
# pargs 'pgrep ttymon'
579: /usr/lib/saf/ttymon -g -h -p system-name console login:
-T sun -d /dev/console -l
argv[0]: /usr/lib/saf/ttymon
argv[1]: -g
argv[2]: -h
argv[3]: -p
argv[4]: system-name console login:
argv[5]: -T
```

例 2-3 プロセスをデバッグする (pargs) (続き)

```
argv[6]: sun
argv[7]: -d
argv[8]: /dev/console
argv[9]: -l
argv[10]: console
argv[11]: -m
argv[12]: ldterm,ttcompat
548: /usr/lib/saf/ttymon
argv[0]: /usr/lib/saf/ttymon
```

次の例は、`pargs -e` コマンドを使用して、プロセスに関連付けられた環境変数を表示する方法を示します。

```
$ pargs -e 6763
6763: tcsh
envp[0]: DISPLAY=:0.0
```

プロセスクラス情報の管理

次のリストは、システム上で構成可能なプロセススケジューリングクラスを示しています。タイムシェアリングクラスのユーザー優先順位の範囲も示しています。

プロセススケジューリングクラスの種類は次のとおりです。

- 公平配分 (FSS)
- 固定優先順位 (FX)
- システム (SYS)
- 対話型 (IA)
- リアルタイム (RT)
- タイムシェアリング (TS)
 - ユーザーが指定する -60 から +60 までの優先順位の範囲。
 - プロセスの優先順位は、親プロセスから継承されます。これを「ユーザーモードの優先順位」と呼びます。
 - システムは、ユーザーモードの優先順位をタイムシェアリングディスパッチパラメータテーブル内で検索します。次にシステムは、`nice` または `pricntl` (ユーザー指定) 優先順位に追加し、0 から 59 までの範囲を確保して「グローバル優先順位」を作成します。

プロセスクラス情報の管理 (タスクマップ)

タスク	説明	参照先
プロセスクラスに関する基本情報を表示する	<code>prionctl -l</code> コマンドを使用し、プロセスのスケジューリングクラスと優先順位の範囲を表示する	40 ページの「プロセスクラスに関する基本情報を表示する方法 (<code>prionctl</code>)」
プロセスのグローバル優先順位を表示する	<code>ps -ecl</code> コマンドを使用し、プロセスのグローバル優先順位を表示する	40 ページの「プロセスのグローバル優先順位を表示する方法」
プロセスに優先順位を割り当てる	<code>prionctl -e -c</code> コマンドを使用し、割り当てた優先順位でプロセスを開始する	42 ページの「プロセスの優先順位を指定する方法 (<code>prionctl</code>)」
タイムシェアリングプロセスのスケジューリングパラメータを変更する	<code>prionctl -s -m</code> コマンドを使用し、タイムシェアリングプロセスのスケジューリングパラメータを変更する	42 ページの「タイムシェアリングプロセスのスケジューリングパラメータを変更する方法 (<code>prionctl</code>)」
プロセスのクラスを変更する	<code>prionctl -s -c</code> コマンドを使用し、プロセスのクラスを変更する	43 ページの「プロセスのクラスを変更する方法 (<code>prionctl</code>)」
プロセスの優先順位を変更する	<code>/usr/bin/nice</code> コマンドを適切なオプションとともに使用し、プロセスの優先順位を低くしたり高くしたりする	45 ページの「プロセスの優先順位を変更する方法 (<code>nice</code>)」

プロセスのスケジュール優先順位の変更 (`prionctl`)

プロセスのスケジュール優先順位とは、スケジュールポリシーに従ってプロセススケジューラによって割り当てられる優先順位のことです。`dispadmin` コマンドを使用すると、デフォルトのスケジュールポリシーを表示できます。詳細は、[dispadmin\(IM\)](#) のマニュアルページを参照してください。

`prionctl` コマンドを使用すると、プロセスを優先順位クラスに割り当てたり、プロセスの優先順位を管理したりできます。`prionctl` コマンドを使用してプロセスを管理する手順については、[42 ページの「プロセスの優先順位を指定する方法 \(`prionctl`\)」](#)を参照してください。

▼ プロセスクラスに関する基本情報を表示する方法 (priocntl)

- **priocntl -l** コマンドを使用して、プロセスのスケジューリングクラスと優先順位の範囲を表示します。

```
$ priocntl -l
```

例 2-4 プロセスクラスに関する基本情報を表示する (priocntl)

次の例に `priocntl -l` コマンドからの出力を示します。

```
# priocntl -l
CONFIGURED CLASSES
=====

SYS (System Class)

TS (Time Sharing)
    Configured TS User Priority Range: -60 through 60

FX (Fixed priority)
    Configured FX User Priority Range: 0 through 60

IA (Interactive)
    Configured IA User Priority Range: -60 through 60
```

▼ プロセスのグローバル優先順位を表示する方法

- **ps** コマンドを使用して、プロセスのグローバル優先順位を表示します。

```
$ ps -ecl
```

グローバル優先順位は、PRI カラムの下に表示されます。

例 2-5 プロセスのグローバル優先順位を表示する

次の例は、`ps -ecl` コマンドの出力を示します。PRI 列の値は、各プロセスの優先順位を示しています。

```
$ ps -ecl
 F S      UID      PID  PPID  CLS  PRI      ADDR      SZ      WCHAN  TTY      TIME CMD
 1 T        0        0    0  SYS  96        ?        0        ?      ?      0:11 sched
 1 S        0        5    0  SDC  99        ?        0        ? ?    ?      0:01 zpool-rp
 0 S        0        1    0   TS  59        ?       688        ? ?    ?      0:00 init
 1 S        0        2    0  SYS  98        ?        0        ? ?    ?      0:00 pageout
 1 S        0        3    0  SYS  60        ?        0        ? ?    ?      2:31 fsflush
 1 S        0        6    0  SDC  99        ?        0        ? ?    ?      0:00 vmtasks
 0 S       16       56    1   TS  59        ?      1026        ? ?    ?      0:01 ipmgmt
```

0 S	0	9	1	TS	59	?	3480	??	0:04	svc.star
0 S	0	11	1	TS	59	?	3480	??	0:13	svc.conf
0 S	0	162	1	TS	59	?	533	??	0:00	pfexecd
0 S	0	1738	1730	TS	59	?	817	? pts/ 1	0:00	bash
0 S	1	852	1	TS	59	?	851	??	0:17	rpcbind
0 S	17	43	1	TS	59	?	1096	??	0:01	netcfgd
0 S	15	47	1	TS	59	?	765	??	0:00	dlnmtd
0 S	0	68	1	TS	59	?	694	??	0:01	in.mpath
0 S	1	1220	1	FX	60	?	682	??	0:00	nfs4cbd
0 S	16	89	1	TS	59	?	1673	??	0:02	nwamd
0 S	0	146	1	TS	59	?	629	??	0:01	dhcpgen
0 S	1	129	1	TS	59	?	1843	??	0:00	kcfcd
0 S	1	1215	1	FX	60	?	738	??	0:00	lockd
0 S	0	829	828	TS	59	?	968	??	0:00	hald-run
0 S	0	361	1	TS	59	?	1081	??	0:01	devfsadm
0 S	0	879	1	TS	59	?	1166	??	0:01	inetd
0 0	119764	1773	880	TS	59	?	557	cons ole	0:00	ps
0 S	0	844	829	TS	59	?	996	??	0:00	hald-add
0 S	0	895	866	TS	59	?	590	??	0:00	ttymon
0 S	0	840	1	TS	59	?	495	??	0:00	cron
0 S	0	874	1	TS	59	?	425	??	0:00	utmpd
0 S	0	1724	956	TS	59	?	2215	??	0:00	sshd
0 S	119764	880	9	TS	59	?	565	? cons ole	0:00	csh
0 S	0	210	1	TS	59	?	1622	??	0:00	sysevent
0 S	0	279	1	TS	59	?	472	??	0:00	iscsid
0 S	1	1221	1	TS	59	?	1349	??	0:00	nfsmapid
1 S	0	374	0	SDC	99	?	0	??	0:00	zpool-us
0 S	0	1207	1	TS	59	?	1063	??	0:00	rmvolmgr
0 S	0	828	1	TS	59	?	1776	??	0:03	hald
0 S	0	853	829	TS	59	?	896	??	0:02	hald-add
0 S	0	373	1	TS	59	?	985	??	0:00	picld
0 S	0	299	1	TS	59	?	836	??	0:00	dbus-dae
0 S	12524	1730	1725	TS	59	?	452	? pts/ 1	0:00	csh
0 S	0	370	1	TS	59	?	574	??	0:00	powerd
0 S	0	264	1	FX	60	?	637	??	0:00	zonestat
0 S	0	866	9	TS	59	?	555	??	0:00	sac
0 S	0	851	829	TS	59	?	998	??	0:00	hald-add
0 S	12524	1725	1724	TS	59	?	2732	??	0:00	sshd
0 S	1	1211	1	TS	59	?	783	??	0:00	statd
0 S	0	1046	1	TS	59	?	1770	??	0:13	intrd
0 S	0	889	1	TS	59	?	1063	??	0:00	syslogd
0 S	0	1209	1	TS	59	?	792	??	0:00	in.ndpd
0 S	0	1188	1186	TS	59	?	951	??	0:15	automoun
0 S	0	1172	829	TS	59	?	725	??	0:00	hald-add
0 S	0	1186	1	TS	59	?	692	??	0:00	automoun
0 S	101	1739	1738	TS	59	?	817	? pts/ 1	0:00	bash
0 S	0	1199	1	TS	59	?	1495	??	0:02	sendmail
0 S	0	956	1	TS	59	?	1729	??	0:00	sshd
0 S	25	1192	1	TS	59	?	1528	??	0:00	sendmail
0 S	0	934	1	TS	59	?	6897	??	0:14	fmd
0 S	0	1131	1	TS	59	?	1691	??	0:07	nscd
0 S	1	1181	1	TS	59	?	699	??	0:00	ypbind

▼ プロセスの優先順位を指定する方法 (priocntl)

- 1 root 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 指定した優先順位でプロセスを起動します。

```
# priocntl -e -c class -m user-limit -p pri command-name
```

-e コマンドを実行します。

-c class プロセスを実行する範囲のクラスを指定する。有効なクラスは TS (タイムシェアリング)、RT (リアルタイム)、IA (対話型)、FSS (公平配分)、および FX (固定優先順位)

-m user-limit このオプションと共に -p オプションを使用すると、優先順位を上下できる最大範囲も指定できる。

-p pri command-name リアルタイムスレッド用に RT クラス内で相対優先順位を指定できるようにする。タイムシェアリングプロセスの場合は、-p オプションを使用すると -60 から +60 までのユーザー指定の優先順位を指定できる

- 3 プロセスステータスを確認します。

```
# ps -ecl | grep command-name
```

例 2-6 プロセスの優先順位を指定する (priocntl)

次の例では、ユーザーが指定できる最上位の優先順位を使用して find コマンドを開始します。

```
# priocntl -e -c TS -m 60 -p 60 find . -name core -print  
# ps -ecl | grep find
```

▼ タイムシェアリングプロセスのスケジューリングパラメータを変更する方法 (priocntl)

- 1 root 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 実行中のタイムシェアリングプロセスのスケジューリングパラメータを変更します。

```
# priocntl -s -m user-limit [-p user-priority] -i idtype idlist
```

-s	ユーザー優先順位の範囲について上限を設定し、現在の優先順位を変更する
-m <i>user-limit</i>	-p オプションを使用するときに、優先順位を上下できる最大範囲を指定する
-p <i>user-priority</i>	優先順位を指定する
-i <i>xidtype xidlist</i>	<i>xidtype</i> と <i>xidlist</i> の組み合わせを使用してプロセスを識別する。「 <i>xidtype</i> 」ではプロセスIDやユーザーIDなど、IDのタイプを指定する。「 <i>xidlist</i> 」ではプロセスIDまたはユーザーIDのリストを識別する

- 3 プロセスステータスを確認します。

```
# ps -ecl | grep idlist
```

例 2-7 タイムシェアリングプロセスのスケジューリングパラメータを変更する (priocntl)

次の例では、500 ミリ秒のタイムスライス、クラス RT 内の優先順位 20、グローバル優先順位 120 を指定して、コマンドを実行します。

```
# priocntl -e -c RT -m 500 -p 20 myprog  
# ps -ecl | grep myprog
```

▼ プロセスのクラスを変更する方法 (priocntl)

- 1 (省略可能) root 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 プロセスのクラスを変更する

```
# priocntl -s -c class -i idtype idlist
```

-s	ユーザー優先順位の範囲について上限を設定し、現在の優先順位を変更する
-c <i>class</i>	クラス TS (タイムシェアリング) または RT (リアルタイム) を指定して、プロセスのクラスを変更する

```
-i idtype idlist    xidtype と xidlist の組み合わせを使用してプロセスを識別する。xidtype ではプロセス ID やユーザー ID など、ID のタイプを指定する。「xidlist」ではプロセス ID またはユーザー ID のリストを識別する
```

注- プロセスをリアルタイムプロセスに変更したり、リアルタイムプロセスから変更したりするには、ユーザーは root 役割になるか、リアルタイムシェル内で作業中ではない必要があります。root としてユーザープロセスをリアルタイムクラスに変更すると、そのユーザーは `prionctl -s` コマンドを使用して、リアルタイムのスケジューリングパラメータを変更できなくなります。

3 プロセスステータスを確認します。

```
# ps -ecl | grep idlist
```

例 2-8 プロセスのクラスを変更する (prionctl)

次の例では、ユーザー 15249 が所有するすべてのプロセスをリアルタイムプロセスに変更します。

```
# prionctl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

タイムシェアリングプロセスの優先順位の変更 (nice)

`nice` コマンドは、旧バージョンとの下位互換性を保つためのみサポートされます。`prionctl` コマンドを使用する方がプロセスを柔軟に管理できます。

プロセスの優先順位は、そのスケジューリングポリシーと `nice` 値によって決定されます。各タイムシェアリングプロセスは、グローバル優先順位を持っています。グローバル優先順位は、ユーザーが指定した優先順位 (`nice` コマンドまたは `prionctl` コマンドの影響を受ける) とシステムで計算された優先順位を加算して算出されます。

プロセスの実行優先順位番号は、オペレーティングシステムによって割り当てられます。優先順位番号は、プロセスのスケジューリングクラス、使用される CPU 時間、`nice` 値 (タイムシェアリングプロセスの場合) などの、複数の要素によって決定されます。

各タイムシェアリングプロセスは、親プロセスから継承したデフォルトの `nice` 値で起動します。`nice` 値は、`ps` レポートの NI カラムに表示されます。

ユーザーは、自分が与える nice 値優先順位を大きくしてプロセスの優先順位を下げるすることができます。ただし、nice 値を小さくしてプロセスの優先順位を上げることができるのは、スーパーユーザーだけです。これは、ユーザーが各自のプロセスの優先順位を大きくして CPU の独占比率を高めるのを防ぐためです。

nice 値の範囲は 0 から +39 までで、0 が最上位の優先順位です。各タイムシェアリングプロセスのデフォルトの nice 値は 20 です。このコマンドには、利用できるバージョンが 2 つあります。標準バージョンの `/usr/bin/nice` と、C シェルの組み込みコマンドです。

▼ プロセスの優先順位を変更する方法 (nice)

この方法により、ユーザーがプロセスの優先順位を下げるすることができます。ただし、root 役割はプロセスの優先順位を上げることも下げることもできます。

- 1 プロセスの優先順位をユーザーとして変更するか、スーパーユーザーとして変更するかを決定します。次のいずれかの手順に従います。

- ユーザーとして、手順 2 の例に従ってコマンドの優先順位を下げます。
- スーパーユーザーとして、手順 3 の例に従ってコマンドの優先順位を上げたり下げたりします。

- 2 ユーザーとして、nice 値を大きくすることでコマンドの優先順位を下げます。

次の nice コマンドは、nice 値を 5 単位分大きくすることで、*command-name* を実行する優先順位を下げます。

```
$ /usr/bin/nice -5 command-name
```

上記のコマンドでは、マイナス記号は次にくるものがオプションであることを表します。このコマンドは、次のように指定することもできます。

```
$ /usr/bin/nice -n 5 command-name
```

次の nice コマンドは、nice 値をデフォルトの 10 単位分大きくすることで、*command-name* の優先順位を下げます。ただし、最大値の 39 を超えさせることはできません。

```
$ /usr/bin/nice command-name
```

- 3 スーパーユーザーとして、**nice** 値を変更してコマンドの優先順位を上げたり下げたりします。

次の **nice** コマンドは、**nice** 値を 10 単位分小さくすることで、*command-name* の優先順位を上げます。ただし、最低値の 0 未満にすることはできません。

```
# /usr/bin/nice --10 command-name
```

上記のコマンドでは、最初のマイナス記号は次にくるものがオプションであることを表します。2 番目のマイナス記号は負の数を表します。

次の **nice** コマンドは、**nice** 値を 5 単位分大きくすることで、*command-name* の優先順位を下げます。ただし、最高値の 39 を超えさせることはできません。

```
# /usr/bin/nice -5 command-name
```

参照 詳細は、[nice\(1\)](#) のマニュアルページを参照してください。

システムのプロセスに関するトラブルシューティング方法

よくある問題に関するヒントを次に示します。

- 同じユーザーが所有する複数の同じジョブがないかどうかを調べます。ジョブが終了するまで待たずに多数のバックグラウンドジョブを起動するスクリプトを実行した場合に、この問題が発生することがあります。
- CPU 時間が大量に増えているプロセスがないかどうかを調べます。この問題を調べるには、**ps** 出力の **TIME** フィールドを確認します。そのプロセスが無限ループに入っている可能性があります。
- 実行中のプロセスの優先順位が高すぎないかどうかを調べます。**ps -c** コマンドを使用して **cls** フィールドを調べると、各プロセスのスケジューラクラスが表示されます。リアルタイム (RT) プロセスとして実行中のプロセスが CPU を独占している可能性があります。また、**nice** 値の高いタイムシェアリング (TS) プロセスがないかどうかを調べます。スーパーユーザー特権を持つユーザーが、プロセスの優先順位を上げすぎた可能性があります。システム管理者は、**nice** コマンドを使用して優先順位を下げることができます。
- 制御がきかなくなったプロセスを調べます。このようなプロセスは、CPU 時間の使用が継続的に増加していきます。プロセスの開始時間 (STIME) と、その後の CPU 時間 (TIME) の累積を調べると、この問題を特定できます。

システムパフォーマンスの監視(タスク)

コンピュータやネットワークのパフォーマンスを十分に引き出すことは、システム管理における重要な作業です。この章では、コンピュータシステムのパフォーマンスの管理に影響する要素について簡単に説明します。また、この章では `vmstat`、`iostat`、`df`、および `sar` の各コマンドを使用してシステムパフォーマンスを監視する手順についても説明します。

この章の内容は次のとおりです:

- 47 ページの「システムパフォーマンスタスクに関する情報の参照箇所」
- 48 ページの「システムパフォーマンスおよびシステムリソース」
- 48 ページの「プロセスとシステムのパフォーマンス」
- 50 ページの「システムパフォーマンスの監視」
- 51 ページの「システムパフォーマンス情報の表示」
- 60 ページの「システム動作の監視」

システムパフォーマンスタスクに関する情報の参照箇所

システムパフォーマンスのタスク	参照先
プロセスの管理	第2章「システムプロセスの管理(タスク)」
システムパフォーマンスの監視	第3章「システムパフォーマンスの監視(タスク)」
調整可能パラメータの変更	『Oracle Solaris 11.1 カーネルのチューンアップ・リファレンスマニュアル』
システムパフォーマンスタスクの管理	『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』の第2章「プロジェクトとタスク(概要)」

システムパフォーマンスのタスク	参照先
FX および FS スケジューラを使用したプロセス管理	『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』の第 8 章「公平配分スケジューラ (概要)」

システムパフォーマンスおよびシステムリソース

コンピュータシステムのパフォーマンスは、システムがリソースをどのように使用し、割り当てるかによって変わります。したがって、通常の条件下でどのように動作するかを知るために、システムパフォーマンスを定期的に監視する必要があります。期待できるパフォーマンスについてよく把握し、問題が発生したときに分析できなければなりません。

パフォーマンスに影響を及ぼすシステムリソースは次のとおりです。

システムリソース	説明
中央処理装置 (CPU)	CPU は、命令をメモリーからフェッチして実行します。
入出力 (I/O) デバイス	I/O デバイスは、コンピュータとの間で情報をやりとりします。この種のデバイスには、端末とキーボード、ディスクドライブ、プリンタなどがあります。
メモリー	物理 (またはメイン) メモリーは、システム上のランダムアクセスメモリー (RAM) の容量を示します。

第 3 章「システムパフォーマンスの監視 (タスク)」では、システムの動作とパフォーマンスに関する統計を表示するツールについて説明します。

プロセスとシステムのパフォーマンス

次の表に、プロセスに関連する用語を示します。

表 3-1 プロセスに関連する用語

用語	説明
プロセス	システムの動作またはジョブ。システムをブートしてコマンドを実行するか、アプリケーションを起動するたびに、システムは 1 つ以上のプロセスをアクティブにする

表 3-1 プロセスに関連する用語 (続き)

用語	説明
軽量プロセス (LWP)	仮想 CPU または実行リソース。LWP は、利用できる CPU リソースをスケジューラクラスと優先順位に基づいて使用するよう に、カーネルによってスケジューラされる。LWP には、カーネルスレッドと LWP が含まれる。カーネルスレッドには、メモリーに常駐する情報が入っている。また、LWP には、スワップ可能な情報が入っている
アプリケーションスレッド	ユーザーのアドレス空間内で独立して実行できる別個のスタックを持った一連の命令。アプリケーションスレッドは LWP の最上部で多重化できる

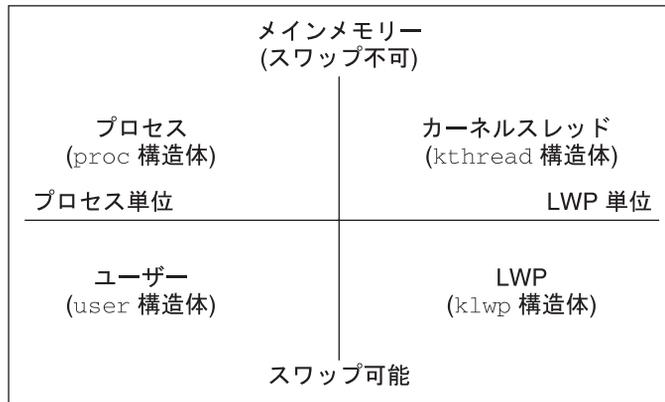
1 つのプロセスは、複数の LWP と複数のアプリケーションスレッドで構成できます。カーネルはカーネルスレッド構造をスケジューラします。この構造は、SunOS 環境内をスケジューラする実体です。次の表に各種プロセス構造体を示します。

表 3-2 プロセス構造体

構造体	説明
proc	プロセス全体に関連し、メインメモリーに常駐しなければならない情報が入っている
kthread	1 つの LWP に関連し、メインメモリーに常駐しなければならない情報が入っている
user	スワップ可能な、プロセス単位の情報が入っている
klwp	スワップ可能な、LWP プロセス単位の情報が入っている

次の図に、これらのプロセス構造体の関係を示します。

図 3-1 プロセス構造体の関係



プロセス内のすべてのスレッドは、ほとんどのプロセスリソースにアクセスできません。ほとんどすべてのプロセスの仮想メモリーが共有されます。あるスレッドが共有データを変更すると、その変更結果をプロセス内の他のスレッドが利用できません。

システムパフォーマンスの監視

コンピュータの稼働中は、各種のシステム動作を追跡するためにオペレーティングシステムのカウンタが増分されます。

追跡されるシステム動作は次のとおりです。

- 中央処理装置 (CPU) の使用状況
- バッファの使用状況
- ディスクとテープの入出力 (I/O) 動作
- 端末デバイスの動作
- システムコールの動作
- コンテキスト切替え
- ファイルアクセス
- 待ち行列の動作
- カーネルテーブル
- プロセス間通信
- ページング
- 空きメモリーとスワップ空間
- カーネルメモリー割り当て (KMA)

監視ツール

Oracle Solaris ソフトウェアには、システムパフォーマンスを追跡できるように複数のツールが提供されています。

表 3-3 パフォーマンス監視ツール

コマンド	説明	参照先
cpustat と cputrack コマンド	CPU パフォーマンスカウンタを使用し、システムのパフォーマンスまたはプロセスを監視する	cpustat(1M) および cputrack(1)
netstat コマンドと nfsstat コマンド	ネットワークパフォーマンスについての情報を表示する	netstat(1M) および nfsstat(1M)
ps コマンドと prstat コマンド	活動中のコマンドについての情報を表示する	第 2 章「システムプロセスの管理 (タスク)」
sar コマンドと sadc コマンド	システム動作データを収集および報告する	第 3 章「システムパフォーマンスの監視 (タスク)」
swap コマンド	ユーザーのシステムで利用可能なスワップ領域についての情報を表示する	『Oracle Solaris 11.1 の管理: デバイスとファイルシステム』の第 16 章「追加スワップ空間の構成 (タスク)」
vmstat コマンドと iostat コマンド	システム動作データの要約。仮想メモリーの統計、ディスクの使用率、CPU の動作など	第 3 章「システムパフォーマンスの監視 (タスク)」
kstat コマンドと mpstat コマンド	システムで使用可能なカーネル統計 (kstats) を検査し、コマンド行で指定された基準に一致する統計を報告する。mpstat コマンドは、プロセス統計を表形式で報告する	kstat(1M) および mpstat(1M) のマニュアルページ

システムパフォーマンス情報の表示

このセクションでは、システムパフォーマンス情報の表示のタスクについて説明します。

システムパフォーマンス情報の表示(タスクマップ)

タスク	説明	参照先
仮想メモリーの統計を表示する	vmstat コマンドを使用し、仮想メモリーの統計を収集する	54 ページの「仮想メモリーの統計情報を表示する方法 (vmstat)」
システムイベント情報を表示する	vmstat コマンドと -s オプションを使用し、システムイベント情報を表示する	54 ページの「システムイベント情報を表示する方法 (vmstat -s)」
スワップの統計を表示する	vmstat コマンドと -s オプションを使用し、スワップの統計を表示する	55 ページの「スワップの統計情報を表示する方法 (vmstat -s)」
各デバイス当たりの割り込み数を表示する	vmstat コマンドと -i オプションを使用し、デバイス当たりの割り込み数を表示する	55 ページの「各デバイス当たりの割り込み数を表示する方法 (vmstat -i)」
ディスクの使用状況を表示する	iostat コマンドを使用し、ディスクの入出力の統計を報告する	56 ページの「ディスクの使用状況を表示する方法 (iostat)」
拡張ディスク統計情報を表示する	iostat コマンドと -xtc オプションを使用し、拡張ディスクの統計情報を表示する	57 ページの「拡張ディスク統計情報を表示する方法 (iostat -xtc)」
ディスク容量情報を表示する	df -k コマンドを使用し、ディスク容量情報をキロバイト単位で表示する	58 ページの「ディスク容量情報を表示する方法 (df -k)」

仮想メモリーの統計情報の表示 (vmstat)

vmstat コマンドを使用すると、仮想メモリーの統計情報と、CPU のロード、ページング、コンテキスト切替え数、デバイス割り込み、システムコールなどの、システムイベントに関する情報を表示できます。また、vmstat コマンドを使用すると、スワップ、キャッシュフラッシュ、および割り込みに関する統計情報も表示できます。

表 3-4 vmstat コマンドからの出力

カテゴリ	フィールド名	説明
procs		次の状態を報告します

表 3-4 vmstat コマンドからの出力 (続き)

カテゴリ	フィールド名	説明
	r	ディスパッチ待ち行列内のカーネルスレッド数
	b	リソースを待機中のブロックされたカーネルスレッド数
	w	リソース処理の完了を待機中のスワップアウトされた軽量プロセス数
memory		実メモリーと仮想メモリーの使用状況を表示します
	swap	使用可能なスワップ空間
	free	空きリストのサイズ
page		ページフォルトとページング動作を1秒当たりの単位数として表示します
	re	回収されたページ数
	mf	軽度の障害と重大な障害
	pi	ページインされたキロバイト数
	po	ページアウトされたキロバイト数
	fr	解放されたキロバイト数
	de	最後にスワップインされたプロセスに必要なと予想されるメモリー
	sr	page デーモンによって走査され、現在は使用されていないページ数。sr が 0 以外の値であれば、page デーモンが実行されています。
disk		最高 4 台のディスク上のデータを示す、1 秒当たりのディスク処理数を表示します
faults		トラップ/割り込み率(1秒あたり)を表示します
	in	1秒当たりの割り込み数
	sy	1秒当たりのシステムコール数
	cs	CPU のコンテキスト切替率
cpu		CPU 時間の使用状況を表示します
	us	ユーザー時間
	sy	システム時間
	id	アイドル時間

このコマンドの詳細については、[vmstat\(1M\)](#)のマニュアルページを参照してください。

▼ 仮想メモリの統計情報を表示する方法 (vmstat)

- 秒単位で時間間隔を指定して `vmstat` コマンドを使用すると、仮想メモリの統計情報が収集されます。

```
$ vmstat n
```

`n` は、秒単位で表した報告間隔です。

例 3-1 仮想メモリの統計情報を表示する

次の例に、5 秒間隔で収集された統計情報に関する `vmstat` の表示を示します。

```
$ vmstat 5
kthr      memory          page        disk        faults        cpu
 r  b  w    swap  free  re  mf  pi  po  fr  de  sr  dd  f0  s1  --   in  sy   cs  us  sy  id
 0  0  0  863160 365680  0   3  1  0  0  0  0  0  0  0  0  406 378  209  1  0  99
 0  0  0  765640 208568  0  36  0  0  0  0  0  0  0  0  0  479 4445 1378  3  3  94
 0  0  0  765640 208568  0   0  0  0  0  0  0  0  0  0  0  423  214  235  0  0 100
 0  0  0  765712 208640  0   0  0  0  0  0  0  3  0  0  0  412  158  181  0  0 100
 0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  157  179  0  0 100
 0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  403  153  182  0  0 100
 0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  168  177  0  0 100
 0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  153  178  0  0 100
 0  0  0  765832 208760  0  18  0  0  0  0  0  0  0  0  0  407  165  186  0  0 100
```

▼ システムイベント情報を表示する方法 (vmstat -s)

- `vmstat -s` コマンドを実行すると、システムを前回ブートしたあとに発生したシステムイベントの数が表示されます。

```
$ vmstat -s
 0 swap ins
 0 swap outs
 0 pages swapped in
 0 pages swapped out
522586 total address trans. faults taken
17006 page ins
 25 page outs
23361 pages paged in
 28 pages paged out
45594 total reclaims
45592 reclaims from free list
 0 micro (hat) faults
522586 minor (as) faults
16189 major faults
98241 copy-on-write faults
137280 zero fill page faults
45052 pages examined by the clock daemon
 0 revolutions of the clock hand
```

```

    26 pages freed by the clock daemon
  2857 forks
    78 vforks
   1647 execs
34673885 cpu context switches
65943468 device interrupts
  711250 traps
63957605 system calls
  3523925 total name lookups (cache hits 99%)
    92590 user   cpu
    65952 system cpu
16085832 idle   cpu
    7450 wait   cpu

```

▼ スワップの統計情報を表示する方法 (vmstat -S)

- **vmstat -S** を実行すると、スワップの統計情報が表示されます。

```

$ vmstat -S
kthr      memory          page            disk           faults        cpu
 r  b  w  swap  free  si  so  pi  po  fr  de  sr  dd  f0  s1  --  in  sy  cs  us  sy  id
  0  0  0 862608 364792  0   0  1  0  0  0  0  0  0  0  0  406 394 213  1  0 99

```

スワッピング統計情報のフィールドを次のリストに示します。その他のフィールドの説明については、表 3-4 を参照してください。

si 1秒あたりにスワップされた平均軽量プロセス数
so スワップアウトされた全プロセス数

注 - vmstat コマンドは、si フィールドおよび so フィールドの出力値を切り捨てます。スワップ統計情報の詳細情報を表示するには、sar コマンドを使用してください。

▼ 各デバイス当たりの割り込み数を表示する方法 (vmstat -i)

- **vmstat -i** コマンドを実行すると、デバイス当たりの割り込み数が表示されます。

例 3-2 各デバイス当たりの割り込み数を表示する

次の例は、vmstat -i コマンドからの出力を示します。

```

$ vmstat -i
interrupt      total      rate
-----
clock          52163269    100
esp0            2600077      4
zsc0            25341        0
zsc1            48917        0
cgsixc0         459          0
lec0           400882        0

```

```

fdc0          14      0
bppc0         0      0
audiocs0      0      0
-----
Total        55238959  105

```

ディスク使用状況の表示 (iostat)

iostat コマンドを使用すると、ディスクの入出力に関する統計情報を表示したり、スループット、使用率、待ち行列の長さ、トランザクション率、およびサービス時間の計測結果を表示したりできます。このコマンドの詳細については、[iostat\(1M\)](#) のマニュアルページを参照してください。

▼ ディスクの使用状況を表示する方法 (iostat)

- 秒単位で時間間隔を指定して **iostat** コマンドを使用すると、ディスクの使用状況が表示されます。

```

$ iostat 5
      tty          fd0          sd3          nfs1          nfs31          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
  0   1   0   0  410    3   0  29   0   0   9   3   0  47   4  2  0  94

```

出力の1行目は、今回のブート以降の統計情報を示します。2行目以降は、時間間隔ごとの統計情報を示します。デフォルトでは、端末 (tty)、ディスク (fd と sd)、および CPU (cpu) の統計情報が表示されます。

例 3-3 ディスクの使用状況を表示する

次の例は、5 秒間隔で収集されるディスク統計情報を示します。

```

$ iostat 5
      tty          sd0          sd6          nfs1          nfs49          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
  0   0   1   0  49   0   0   0   0   0   0   0   0   15  0  0  0  100
  0  47   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16  44   6  132   0   0   0   0   0   0   0   0   0   0  0  0  1  99
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   3   1  23   0   0   0   0   0   0   0   0   0   0  0  0  1  99
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100

```

次の表に、`iostat n` コマンド出力内のフィールドを示します。

デバイスの種類	フィールド名	説明
端末	デバイスの種類	
	tin	端末の入力待ち行列内の文字数
	tout	端末の出力待ち行列内の文字数
ディスク	デバイスの種類	
	bps	1秒当たりのブロック数
	tps	1秒当たりのトランザクション数
	serv	ミリ秒単位で表した平均サービス時間
CPU	デバイスの種類	
	us	ユーザーモード
	sy	システムモード
	wt	入出力待機中
	id	アイドル状態

▼ 拡張ディスク統計情報を表示する方法 (iostat -xtc)

- **iostat -xtc** コマンドを実行すると、拡張ディスク統計情報が表示されます。

```
$ iostat -xtc
                extended device statistics
device      r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b   tty      cpu
            tin tout  us sy wt id
fd0          0.0    0.0    0.0    0.0  0.0  0.0    0.0  0  0    0    0  0  0  100
sd0          0.0    0.0    0.4    0.4  0.0  0.0    49.5  0  0
sd6          0.0    0.0    0.0    0.0  0.0  0.0    0.0  0  0
nfs1         0.0    0.0    0.0    0.0  0.0  0.0    0.0  0  0
nfs49        0.0    0.0    0.0    0.0  0.0  0.0    15.1  0  0
nfs53        0.0    0.0    0.4    0.0  0.0  0.0    24.5  0  0
nfs54        0.0    0.0    0.0    0.0  0.0  0.0    6.3  0  0
nfs55        0.0    0.0    0.0    0.0  0.0  0.0    4.9  0  0
```

iostat -xtc コマンドを使用すると、ディスクごとに1行ずつ出力が表示されます。出力フィールドを次のリストに示します。

r/s 1秒当たりの読み取り数
w/s 1秒当たりの書き込み数
kr/s 1秒当たりの読み取りキロバイト数
kw/s 1秒当たりの書き込みキロバイト数
wait サービス(待ち行列の長さ)を待機中の平均トランザクション数

actv	サービス中の平均トランザクション数
svc_t	ミリ秒単位で表した平均サービス時間
%w	待ち行列が空でない時間の割合
%b	ディスクがビジーである時間の割合

ディスク容量統計情報の表示 (df)

df コマンドを使用すると、マウントされている各ディスク上の空きディスク容量が表示されます。レポート用の統計情報では使用可能容量の合計の内先頭に 10% の空き容量を残しておくので、df から報告される「使用可能」ディスク容量は全容量の 90% のみに相当します。この「先頭の空き容量」は、パフォーマンスを高めるために常に空になっています。

実際に df コマンドからレポートされるディスク容量の割合は、使用済み容量を使用可能容量で割った値です。

ファイルシステムの容量が 90% を超える場合、cp コマンドを使用して空いているディスクにファイルを転送できます。また、tar コマンドか cpio コマンドを使用してテープにファイルを転送することもできます。また、ファイルを削除することもできます。

このコマンドの詳細については、[df\(1M\)](#) のマニュアルページを参照してください。

▼ ディスク容量情報を表示する方法 (df -k)

- df -k コマンドを使用すると、ディスク容量情報がキロバイト単位で表示されます。

```
$ df -k
Filesystem          kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0  192807   40231  133296    24%     /
```

例 3-4 ファイルシステム情報を表示する

次の例は、df -k コマンドの出力です。

```
$ df -k
Filesystem          1024-blocks    Used Available Capacity  Mounted on
rpool/ROOT/solaris-161 191987712 6004395 140577816    5%     /
/devices              0           0         0    0%   /devices
/dev                  0           0         0    0%   /dev
ctfs                  0           0         0    0%   /system/contract
proc                  0           0         0    0%   /proc
mnttab                 0           0         0    0%   /etc/mnttab
swap                  4184236     496     4183740    1%   /system/volatile
objfs                  0           0         0    0%   /system/object
```

```

sharefs                0          0          0  0% /etc/dfs/sharetab
/usr/lib/libc/libc_hwcapi.so.1 146582211 6004395 140577816 5% /lib/libc.so.1
fd                    0          0          0 0% /dev/fd
swap                 4183784    60      4183724 1% /tmp
rpool/export         191987712 35      140577816 1% /export
rpool/export/home    191987712 32      140577816 1% /export/home
rpool/export/home/123 191987712 13108813 140577816 9% /export/home/123
rpool/export/repo    191987712 11187204 140577816 8% /export/repo
rpool/export/repo2010_11 191987712 31 140577816 1% /export/repo2010_11
rpool                191987712 5238974 140577816 4% /rpool
/export/home/123     153686630 13108813 140577816 9% /home/123

```

次の表に、df -k コマンドの出力を説明します。

フィールド名	説明
kbytes	ファイルシステム内の使用可能容量の合計
used	使用されている容量
avail	使用可能容量
capacity	使用されている容量が全容量に占める割合
mounted on	マウントポイント

例 3-5 オプションを指定しないdf コマンドを使用してファイルシステム情報を表示する

オペランドやオプションを指定せずにdf コマンドを使用すると、次の例に示すように、マウントされているすべてのファイルシステムが報告されます。

```

$ df
/ (rpool/ROOT/solaris):100715496 blocks 100715496 files
/devices (/devices ): 0 blocks 0 files
/dev (/dev ): 0 blocks 0 files
/system/contract (ctfs ): 0 blocks 2147483601 files
/proc (proc ): 0 blocks 29946 files
/etc/mnttab (mnttab ): 0 blocks 0 files
/system/volatile (swap ):42257568 blocks 2276112 files
/system/object (objfs ): 0 blocks 2147483441 files
/etc/dfs/sharetab (sharefs ): 0 blocks 2147483646 files
/dev/fd (fd ): 0 blocks 0 files
/tmp (swap ):42257568 blocks 2276112 files
/export (rpool/export ):100715496 blocks 100715496 files
/export/home (rpool/export/home ):100715496 blocks 100715496 files
/export/home/admin (rpool/export/home/admin):100715496 blocks 100715496 files
/rpool (rpool ):100715496 blocks 100715496 files
/export/repo2010_11(rpool/export/repo2010_11):281155639 blocks 281155639 files
/rpool (rpool ):281155639 blocks 281155639 files

```

システム動作の監視

このセクションでは、システム動作の監視のタスクについて説明します。

システム動作の監視(タスクマップ)

タスク	説明	参照先
ファイルアクセスをチェックします。	sar コマンドと -a オプションを使用し、ファイルアクセス操作ステータスを表示します。	62 ページの「ファイルアクセスをチェックする方法 (sar -a)」
バッファ動作をチェックします。	sar コマンドと -b オプションを使用し、バッファ動作の統計情報を表示します。	63 ページの「バッファ動作をチェックする方法 (sar -b)」
システムコールの統計情報をチェックします。	sar コマンドと -c オプションを使用し、システムコールの統計情報を表示します。	64 ページの「システムコールの統計情報をチェックする方法 (sar -c)」
ディスク動作をチェックします。	sar コマンドと -d オプションを使用し、ディスク動作をチェックします。	65 ページの「ディスク動作をチェックする方法 (sar -d)」
ページアウトとメモリーをチェックします。	sar コマンドと -g オプションを使用し、ページアウトとメモリー解放動作を表示します。	67 ページの「ページアウトとメモリーをチェックする方法 (sar -g)」
カーネルメモリーの割り当てをチェックします。	カーネルメモリーの割り当て (KMA) では、カーネルサブシステムが必要に応じてメモリーを割り当てたり解放したりします。sar コマンドと -k オプションを使用し、KMA をチェックします。	69 ページの「カーネルメモリーの割り当てをチェックする方法 (sar -k)」
プロセス間通信をチェックします。	sar コマンドと -m オプションを使用し、プロセス間通信の動作を報告します。	70 ページの「プロセス間通信をチェックする方法 (sar -m)」
ページイン動作をチェックします。	sar コマンドと -p オプションを使用し、ページイン動作を報告します。	71 ページの「ページイン動作をチェックする方法 (sar -p)」
待ち行列動作をチェックします。	sar コマンドと -q オプションを使用し、以下をチェックします。 <ul style="list-style-type: none"> ■ 待ち行列に要求が入っている間の平均待ち行列の長さ ■ 待ち行列に要求が入っている時間の割合 	72 ページの「待ち行列動作をチェックする方法 (sar -q)」

タスク	説明	参照先
未使用メモリーをチェックします。	sar コマンドと <code>-r</code> オプションを使用し、現在使用されているメモリーページ数とスワップファイルのディスクブロック数を表示します。	73 ページの「未使用のメモリーをチェックする方法 (sar -r)」
CPU の使用率をチェックします。	sar コマンドと <code>-u</code> オプションを使用し、CPU 使用率を表示します。	74 ページの「CPU の使用状況をチェックする方法 (sar -u)」
システムテーブルのステータスをチェックします。	以下の sar コマンドと <code>-v</code> オプションを使用し、システムテーブルのステータスをチェックします。 <ul style="list-style-type: none"> ■ プロセス ■ i ノード ■ ファイル ■ 共有メモリーレコード 	75 ページの「システムテーブルのステータスをチェックする方法 (sar -v)」
スワップ動作をチェックします。	sar コマンドと <code>-w</code> オプションを使用し、スワップ動作をチェックします。	76 ページの「スワップ動作をチェックする方法 (sar -w)」
端末動作をチェックします。	sar コマンドと <code>-y</code> オプションを使用し、端末デバイスの動作を監視します。	77 ページの「端末動作をチェックする方法 (sar -y)」
システム全体のパフォーマンスをチェックします。	sar <code>-A</code> コマンドを使用し、すべてのオプションを指定した場合と同じように、システム全体のパフォーマンスを示す統計情報を表示します。	78 ページの「システム全体のパフォーマンスをチェックする方法 (sar -A)」
データの自動収集を設定します。	システムでデータを自動的に収集するよう設定し、sar コマンドを実行するには、以下を実行します。 <ul style="list-style-type: none"> ■ <code>svcadm enable system/sar:default</code> コマンドを実行します ■ <code>/var/spool/cron/crontabs/sys</code> ファイルを編集します 	81 ページの「自動データ収集を設定する方法」

システム動作の監視 (sar)

以下のタスクを実行するには、sar コマンドを使用します。

- システム動作についてのデータを編成し表示します。
- 特別な要求に基づいて、システム動作データにアクセスします。

- システムパフォーマンスを測定および監視するレポートを自動的に生成します。また、特定のパフォーマンス障害を正確に突き止めるための、特別な要求レポートも生成します。sar コマンドを設定してシステム上で実行する方法と、これらのツールの説明については、78 ページの「システム動作データの自動収集 (sar)」を参照してください。

このコマンドの詳細については、[sar\(1\)](#) のマニュアルページを参照してください。

▼ ファイルアクセスをチェックする方法 (sar -a)

- **sar -a** コマンドを使用すると、ファイルアクセス操作の統計情報が表示されます。

```
$ sar -a
SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v    ...

00:00:00  iget/s namei/s dirbk/s
01:00:00      0      3      0
02:00:00      0      3      0
03:00:00      0      3      0
04:00:00      0      3      0
05:00:00      0      3      0
06:00:00      0      3      0
07:00:00      0      3      0
08:00:00      0      3      0
08:20:01      0      3      0
08:40:00      0      3      0
09:00:00      0      3      0
09:20:01      0     10      0
09:40:01      0      1      0
10:00:02      0      5      0

Average      0      4      0
```

次に、sar -a コマンドによって表示される、フィールド名とオペレーティングシステムのルーチンを示します。

iget/s ディレクトリ名検索キャッシュ (DNLC) 内に入っていない i ノードに対して出された要求数

namei/s 1 秒当たりのファイルシステムパスの検索数。namei で DNLC 内にディレクトリ名が見つからない場合は、iget が呼び出され、ファイルまたはディレクトリの i ノードが取得される。したがって、ほとんどの igets は DNLC が欠落した結果である

dirbk/s 1 秒間に実行されたディレクトリブロックの読み取り回数

これらのオペレーティングシステムルーチンに対して表示される値が大きいほど、カーネルはユーザーファイルへのアクセスに長い時間を費やしています。この時間には、プログラムとアプリケーションによるファイルシステムの使用量が反映されます。-a オプションを使用すると、アプリケーションのディスク依存度を表示できるので便利です。

▼ バッファ動作をチェックする方法 (sar -b)

- **sar -b** コマンドを使用すると、バッファ動作の統計情報が表示されます。バッファは、メタデータをキャッシュするために使用されます。メタデータには、iノード、シリンダグループブロック、間接ブロックなどがあります。

```
$ sar -b
00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0    100      0      0     55      0      0
```

例 3-6 バッファ動作をチェックする (sar -b)

次の **sar -b** コマンド出力の例は、**%rcache** バッファと **%wcache** バッファが処理速度を低下させていないことを示します。すべてのデータは許容範囲に収まっています。

```
$ sar -b
SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v ...

00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0    100      0      0     94      0      0
02:00:01      0      0    100      0      0     94      0      0
03:00:00      0      0    100      0      0     92      0      0
04:00:00      0      1    100      0      1     94      0      0
05:00:00      0      0    100      0      0     93      0      0
06:00:00      0      0    100      0      0     93      0      0
07:00:00      0      0    100      0      0     93      0      0
08:00:00      0      0    100      0      0     93      0      0
08:20:00      0      1    100      0      1     94      0      0
08:40:01      0      1    100      0      1     93      0      0
09:00:00      0      1    100      0      1     93      0      0
09:20:00      0      1    100      0      1     93      0      0
09:40:00      0      2    100      0      1     89      0      0
10:00:00      0      9    100      0      5     92      0      0
10:20:00      0      0    100      0      0     68      0      0
10:40:00      0      1     98      0      1     70      0      0
11:00:00      0      1    100      0      1     75      0      0

Average      0      1    100      0      1     91      0      0
```

次の表は、**-b** オプションを指定したときに表示されるバッファ動作を示します。

フィールド名	説明
bread/s	ディスクからバッファキャッシュに投入された1秒当たりの平均読み取り数
lread/s	バッファキャッシュからの1秒当たりの平均論理読み取り数

フィールド名	説明
%rcache	バッファークャッシュ内で見つかった論理読み込み数の割合 (lread/s に対する bread/s の比を 100% から差し引いた値)
bwrit/s	バッファークャッシュからディスクに書き込まれた 1 秒当たりの平均物理ブロック数 (512 バイト)
lwrit/s	バッファークャッシュへの 1 秒当たりの平均論理書き込み数
%wcache	バッファークャッシュ内で見つかった論理書き込み数の割合 (lwrit/s に対する bwrit/s の比を 100% から差し引いた値)
pread/s	キャラクタ型デバイスインタフェースを使用する 1 秒当たりの平均物理読み取り数
pwrit/s	キャラクタ型デバイスインタフェースを使用する 1 秒当たりの平均物理書き込み要求数

もっとも重要なエントリは、キャッシュヒット率 %rcache と %wcache です。これらのエントリは、システムバッファリングの有効性を測定します。%rcache が 90% 未満の場合や、%wcache が 65% 未満の場合は、バッファ領域を大きくすればパフォーマンスを改善できる可能性があります。

▼ システムコールの統計情報をチェックする方法 (sar -c)

- sar -c コマンドを使用すると、システムコールの統計情報が表示されます。

```
$ sar -c
00:00:00 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
01:00:00      38      2      2  0.00  0.00   149   120
```

例 3-7 システムコールの統計情報をチェックする (sar -c)

次の例は、sar -c コマンドからの出力を示します。

```
$ sar -c
SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
01:00:00      89      14      9  0.01  0.00   2906   2394
02:00:01      89      14      9  0.01  0.00   2905   2393
03:00:00      89      14      9  0.01  0.00   2908   2393
04:00:00      90      14      9  0.01  0.00   2912   2393
05:00:00      89      14      9  0.01  0.00   2905   2393
06:00:00      89      14      9  0.01  0.00   2905   2393
07:00:00      89      14      9  0.01  0.00   2905   2393
08:00:00      89      14      9  0.01  0.00   2906   2393
08:20:00      90      14      9  0.01  0.01   2914   2395
08:40:01      90      14      9  0.01  0.00   2914   2396
09:00:00      90      14      9  0.01  0.01   2915   2396
09:20:00      90      14      9  0.01  0.01   2915   2396
```

09:40:00	880	207	156	0.08	0.08	26671	9290
10:00:00	2020	530	322	0.14	0.13	57675	36393
10:20:00	853	129	75	0.02	0.01	10500	8594
10:40:00	2061	524	450	0.08	0.08	579217	567072
11:00:00	1658	404	350	0.07	0.06	1152916	1144203
Average	302	66	49	0.02	0.01	57842	55544

次の表は、`-c` オプションを指定したときに報告されるシステムコールのカテゴリを示します。一般に、読み取りと書き込みがシステムコール合計の約半分を占めます。ただし、システムで実行中の動作によってこの割合は大幅に変動します。

フィールド名	説明
<code>scall/s</code>	1秒当たりのすべてのタイプのシステムコール数(通常は、4ユーザーから6ユーザーのシステム上で1秒当たり約30)
<code>sread/s</code>	1秒当たりの <code>read</code> システムコール数
<code>swrit/s</code>	1秒当たりの <code>write</code> システムコール数
<code>fork/s</code>	1秒当たりの <code>fork</code> システムコール数(4ユーザーから6ユーザーのシステム上で毎秒約0.5)。この数値は、シェルスクリプトの実行中は大きくなる
<code>exec/s</code>	1秒当たりの <code>exec</code> システムコール数。 <code>exec/s</code> を <code>fork/s</code> で割った値が3より大きい場合は、効率の悪い <code>PATH</code> 変数を調べる
<code>rchar/s</code>	<code>read</code> システムコールによって転送される1秒当たりの文字(バイト)数
<code>wchar/s</code>	<code>write</code> システムコールによって転送される1秒当たりの文字(バイト)数

▼ ディスク動作をチェックする方法 (`sar -d`)

- `sar -d` コマンドを使用すると、ディスク動作の統計情報が表示されます。

```
$ sar -d
00:00:00 device          %busy  avque  r+w/s  blks/s  await  avserv
```

例3-8 ディスク動作をチェックする

次の例は、`sar -d` コマンドからの一部省略した出力を示します。

```
$ sar -d
Sun05 balmy 5.10 Generic_144500-10 sun4v  ...
12:36:32 device          %busy  avque  r+w/s  blks/s  await  avserv
```

12:40:01	dad1	15	0.7	26	399	18.1	10.0
	dad1,a	15	0.7	26	398	18.1	10.0
	dad1,b	0	0.0	0	1	1.0	3.0
	dad1,c	0	0.0	0	0	0.0	0.0
	dad1,h	0	0.0	0	0	0.0	6.0
	fd0	0	0.0	0	0	0.0	0.0
	nfs1	0	0.0	0	0	0.0	0.0
	nfs2	1	0.0	1	12	0.0	13.2
	nfs3	0	0.0	0	2	0.0	1.9
	nfs4	0	0.0	0	0	0.0	7.0
	nfs5	0	0.0	0	0	0.0	57.1
	nfs6	1	0.0	6	125	4.3	3.2
	nfs7	0	0.0	0	0	0.0	6.0
	sd1	0	0.0	0	0	0.0	5.4
	ohci0,bu	0	0.0	0	0	0.0	0.0
	ohci0,ct	0	0.0	0	0	0.0	0.0
	ohci0,in	0	0.0	7	0	0.0	0.0
	ohci0,is	0	0.0	0	0	0.0	0.0
	ohci0,to	0	0.0	7	0	0.0	0.0

次の表は、-d オプションを指定したときに報告されるディスクデバイスの動作を示します。

フィールド名	説明
デバイス	監視中のディスクデバイス名
%busy	デバイスが転送要求のサービスでビジーであった時間
avque	デバイスが転送要求のサービスでビジーであった時間の間の平均要求数
r+w/s	デバイスへの1秒当たりの読み取り転送数と書き込み転送数
blks/s	デバイスに転送される1秒当たりの512バイトブロック数
await	待ち行列内の転送要求の平均アイドル時間(ミリ秒単位)。これは、待ち行列が占有されている場合のみ計測される
avserv	デバイスが1転送要求の処理に要する平均時間(ミリ秒単位)。ディスクの場合は、この値にシークタイム、回転待ち時間、およびデータ転送時間が含まれる

待ち行列内に何かがあるときは、待ち行列の長さで待ち時間が計測されるので注意してください。%busy の値が小さい場合に、待ち行列とサービス時間が大きければ、変更されたブロックをディスクに随時書き込むために、システムが定期的に処理していることを示す場合があります。

▼ ページアウトとメモリーをチェックする方法 (sar -g)

- **sar -g** コマンドを使用すると、ページアウト動作の平均値とメモリー解放動作の平均値とが表示されます。

```
$ sar -g
00:00:00  pgout/s  ppgout/s  pgfree/s  pgscan/s  %ufs_ipf
01:00:00    0.00    0.00    0.00    0.00    0.00
```

sar -g コマンドで表示される出力は、より多くのメモリーが必要かどうかを判断するのに役立ちます。ps -elf コマンドを使用すると、page デーモンに使用される CPU サイクル数が表示されます。サイクル数が大きく、pgfree/s フィールドと pgscan/s フィールドの値が大きければ、メモリー不足を示します。

また、sar -g を使用すると、i ノードの再利用間隔が短すぎるために、再利用可能なページが失われているのかも表示されます。

例 3-9 ページアウトとメモリーをチェックする (sar -g)

次の例は、sar -g コマンドからの出力を示します。

```
$ sar -g
SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:00  pgout/s  ppgout/s  pgfree/s  pgscan/s  %ufs_ipf
01:00:00    0.00    0.00    0.00    0.00    0.00
02:00:00    0.01    0.01    0.01    0.00    0.00
03:00:00    0.00    0.00    0.00    0.00    0.00
04:00:00    0.00    0.00    0.00    0.00    0.00
05:00:00    0.00    0.00    0.00    0.00    0.00
06:00:00    0.00    0.00    0.00    0.00    0.00
07:00:00    0.00    0.00    0.00    0.00    0.00
08:00:00    0.00    0.00    0.00    0.00    0.00
08:20:01    0.00    0.00    0.00    0.00    0.00
08:40:00    0.00    0.00    0.00    0.00    0.00
09:00:00    0.00    0.00    0.00    0.00    0.00
09:20:01    0.05    0.52    1.62    10.16    0.00
09:40:01    0.03    0.44    1.47    4.77    0.00
10:00:02    0.13    2.00    4.38    12.28    0.00
10:20:03    0.37    4.68    12.26    33.80    0.00

Average    0.02    0.25    0.64    1.97    0.00
```

次の表に -g オプションからの出力について説明します。

フィールド名	説明
pgout/s	1 秒間にページアウトされた要求数
ppgout/s	1 秒間に実際にページアウトされたページ数。1つのページアウト要求で複数のページがページアウトされることがあります。

フィールド名	説明
pgfree/s	空きリストに配置された1秒当たりのページ数。
pgscan/s	page デーモンによって走査された1秒当たりのページ数。この値が大きい場合は、page デーモンが空きメモリのチェックに大量の時間を費やしています。これは、メモリーを増やす必要があることを示します。
%ufs_ipf	ufs の i ノードがそれに関連付けられた再使用可能ページを持つ iget によって、空きリストから取り出された割合。これらのページはフラッシュされ、プロセスが回収できなくなります。したがって、このフィールドはページフラッシュを伴う igets の割合です。値が大きければ、i ノードの空きリストがページ境界であり、ufs の i ノード数を増やす必要があることを示します。

カーネルメモリーの割り当て (KMA) のチェック

KMA を使用すると、カーネルサブシステムが必要に応じてメモリーを割り当てたり、解放したりできます。

KMA では、ロードのピーク時に必要と思われる最大メモリー容量を静的に割り当てることをせず、メモリーの要求を次の3つのカテゴリに分けます。

- 「小」 (256 バイト未満)
- 「大」 (512 バイト - 4K バイト)
- 「サイズ超過」 (4K バイト超)

KMA は、2つのメモリープールを管理して、「小」要求と「大」要求を満たします。「サイズ超過」要求は、システムページアロケータからメモリーを割り当てることで満たされます。

KMA リソースを使用するドライバや STREAMS の作成に使用中のシステムを調査する場合は、sar -k コマンドを使用すると便利です。それ以外の場合は、このコマンドで提供される情報は不要です。KMA リソースを使用するが、終了前には特にリソースを返さないドライバやモジュールがあると、メモリーのリークが生じることがあります。メモリーリークが発生すると、KMA によって割り当てられるメモリーは時間が経つにつれて増大します。したがって、sar -k コマンドの alloc フィールドの値が時間が経つにつれ増える場合は、メモリーリークの可能性があります。メモリーリークのもう1つの兆候は、要求が失敗することです。この問題が発生した場合は、メモリーリークのために KMA がメモリーを予約したり割り当てたりできなくなっている可能性があります。

メモリーリークが発生した場合は、KMA からメモリーを要求したが返していないドライバや STREAMS がないかどうかをチェックする必要があります。

▼ カーネルメモリーの割り当てをチェックする方法 (sar -k)

- sar -k コマンドを使用すると、KMA に関して次の動作が表示されます。

```
$ sar -k
00:00:00 sml_mem alloc fail lg_mem alloc fail ovsz_alloc fail
01:00:00 2523136 1866512 0 18939904 14762364 0 360448 0
02:00:02 2523136 1861724 0 18939904 14778748 0 360448 0
```

例 3-10 カーネルメモリーの割り当てをチェックする(sar -k)

次の例は、sar -k 出力を示します。

```
$ sar -k

SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 sml_mem alloc fail lg_mem alloc fail ovsz_alloc fail
01:00:00 6119744 4852865 0 60243968 54334808 156 9666560 0
02:00:01 6119744 4853057 0 60243968 54336088 156 9666560 0
03:00:00 6119744 4853297 0 60243968 54335760 156 9666560 0
04:00:00 6119744 4857673 0 60252160 54375280 156 9666560 0
05:00:00 6119744 4858097 0 60252160 54376240 156 9666560 0
06:00:00 6119744 4858289 0 60252160 54375608 156 9666560 0
07:00:00 6119744 4858793 0 60252160 54442424 156 9666560 0
08:00:00 6119744 4858985 0 60252160 54474552 156 9666560 0
08:20:00 6119744 4858169 0 60252160 54377400 156 9666560 0
08:40:01 6119744 4857345 0 60252160 54376880 156 9666560 0
09:00:00 6119744 4859433 0 60252160 54539752 156 9666560 0
09:20:00 6119744 4858633 0 60252160 54410920 156 9666560 0
09:40:00 6127936 5262064 0 60530688 55619816 156 9666560 0
10:00:00 6545728 5823137 0 62996480 58391136 156 9666560 0
10:20:00 6545728 5758997 0 62996480 57907400 156 9666560 0
10:40:00 6734144 6035759 0 64389120 59743064 156 10493952 0
11:00:00 6996288 6394872 0 65437696 60935936 156 10493952 0

Average 6258044 5150556 0 61138340 55609004 156 9763900 0
```

次の表に -k オプションからの出力について説明します。

フィールド名	説明
sml_mem	KMA が小メモリー要求プール内で使用できるメモリーのバイト数(このプールでは、小要求は 256 バイト未満)
alloc	KMA が小メモリー要求プールから小メモリー要求に割り当てたメモリーのバイト数
fail	失敗した小メモリー要求数
lg_mem	KMA が大メモリー要求プール内で使用できるメモリーのバイト数(このプールでは、大要求は 512 バイトから 4K バイトまで)

フィールド名	説明
alloc	KMAが大メモリー要求プールから大メモリー要求に割り当てたメモリーのバイト数
fail	失敗した大メモリー要求数
ovsz_alloc	サイズ超過要求(4Kバイトを超える要求)に割り当てられたメモリーの容量。これらの要求はページアロケータによって満たされる。このため、プールはない
fail	失敗したサイズ超過メモリー要求数

▼ プロセス間通信をチェックする方法 (sar -m)

- **sar -m** コマンドを使用すると、プロセス間通信の動作が表示されます。

```
$ sar -m
00:00:00  msg/s  sema/s
01:00:00  0.00   0.00
```

通常、これらの数字は、メッセージやセマフォーを使用するアプリケーションを実行していない限りゼロ (0.00) です。

次の表に -m オプションからの出力について説明します。

```
msg/s    1秒当たりのメッセージ処理(送受信)数
sema/s   1秒当たりのセマフォー処理数
```

例 3-11 プロセス間通信をチェックする (sar -m)

次の例は、sar -m コマンドからの一部省略した出力を示します。

```
$ sar -m
SunOS balmy 5.10 Generic_144500-10 sun4v  ...

00:00:00  msg/s  sema/s
01:00:00  0.00   0.00
02:00:02  0.00   0.00
03:00:00  0.00   0.00
04:00:00  0.00   0.00
05:00:01  0.00   0.00
06:00:00  0.00   0.00

Average   0.00   0.00
```

▼ ページイン動作をチェックする方法 (sar -p)

- **sar -p** コマンドを使用すると、保護フォルトや変換フォルトを含むページイン動作が表示されます。

```
$ sar -p
00:00:00 atch/s pgin/s ppgin/s pflt/s vflt/s slock/s
01:00:00 0.07 0.00 0.00 0.21 0.39 0.00
```

例 3-12 ページイン動作をチェックする (sar -p)

次の例は、sar -p コマンドからの出力を示します。

```
$ sar -p
SunOS balmy 5.10 Generic_144500-10 sun4v ...

00:00:04 atch/s pgin/s ppgin/s pflt/s vflt/s slock/s
01:00:00 0.09 0.00 0.00 0.78 2.02 0.00
02:00:01 0.08 0.00 0.00 0.78 2.02 0.00
03:00:00 0.09 0.00 0.00 0.81 2.07 0.00
04:00:00 0.11 0.01 0.01 0.86 2.18 0.00
05:00:00 0.08 0.00 0.00 0.78 2.02 0.00
06:00:00 0.09 0.00 0.00 0.78 2.02 0.00
07:00:00 0.08 0.00 0.00 0.78 2.02 0.00
08:00:00 0.09 0.00 0.00 0.78 2.02 0.00
08:20:00 0.11 0.00 0.00 0.87 2.24 0.00
08:40:01 0.13 0.00 0.00 0.90 2.29 0.00
09:00:00 0.11 0.00 0.00 0.88 2.24 0.00
09:20:00 0.10 0.00 0.00 0.88 2.24 0.00
09:40:00 2.91 1.80 2.38 4.61 17.62 0.00
10:00:00 2.74 2.03 3.08 8.17 21.76 0.00
10:20:00 0.16 0.04 0.04 1.92 2.96 0.00
10:40:00 2.10 2.50 3.42 6.62 16.51 0.00
11:00:00 3.36 0.87 1.35 3.92 15.12 0.00

Average 0.42 0.22 0.31 1.45 4.00 0.00
```

次の表に -p オプションから報告される統計情報を示します。

フィールド名	説明
atch/s	現在メモリーに入っているページを回収して満たされる1秒当たりのページフォルト数(1秒当たりの付加数)。この例には、空きリストから無効なページを回収し、別のプロセスに現在使用中のテキストページを共有する処理が含まれます。たとえば、複数のプロセスが同じプログラムテキストにアクセスしている場合などです。
pgin/s	ファイルシステムがページイン要求を受信する1秒当たりの回数。

フィールド名	説明
ppgin/s	ページインされる1秒当たりのページ数。ソフトロック要求 (slock/s を参照) などの1つのページイン要求や、大型ブロックサイズでは、複数のページがページインされることがあります。
pflt/s	保護エラーによるページフォルト数。保護フォルトの例には、ページへの不正なアクセスや、「書き込み時コピー」などがあります。通常、この数値は主に「書き込み時コピー」からなっています。
vflt/s	1秒当たりのアドレス変換ページフォルト数。このフォルトは、有効性フォルトと呼ばれます。有効性フォルトは、仮想アドレスに有効なプロセステーブルエントリが存在しないときに発生します。
slock/s	物理入出力を要求するソフトウェアロック要求によって発生する1秒当たりのフォルト数。ソフトロック要求の発生例には、ディスクからメモリーへのデータ転送などがあります。システムはデータを受信するページをロックするので、別のプロセスはそれを回収して使用できません。

▼ 待ち行列動作をチェックする方法 (sar -q)

- **sar -q** コマンドを使用すると、以下の情報を報告できます。
 - 待ち行列に要求が入っている間の平均待ち行列の長さ
 - 待ち行列に要求が入っている時間の割合

```
$ sar -q
00:00:00 runq-sz %runocc swpq-sz %swpocc
```

次の表に -q オプションからの出力について説明します。

runq-sz	CPU を実行するためにメモリー内で待機中のカーネルスレッド数。通常、この値は2未満になる。値が常に2より大きい場合は、システムがCPUの限界に到達している可能性がある
%runocc	ディスク待ち行列が使用されている時間の割合
swpq-sz	スワップアウトされるプロセスの平均数。
%swpocc	プロセスがスワップアウトされている時間の割合。

例 3-13 待ち行列動作をチェックする

次の例は、`sar -q` コマンドからの出力を示します。`%runocc` の値が大きく (90 パーセント超)、`runq-sz` の値が 2 より大きい場合は、CPU の負荷が大きく、応答速度が低下しています。この場合は、CPU の容量を増やしてシステムの応答速度を適正化する必要があります。

```
# sar -q
SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00    1.0      7      0.0      0
02:00:00    1.0      7      0.0      0
03:00:00    1.0      7      0.0      0
04:00:00    1.0      7      0.0      0
05:00:00    1.0      6      0.0      0
06:00:00    1.0      7      0.0      0

Average      1.0      7      0.0      0
```

▼ 未使用のメモリーをチェックする方法 (sar -r)

- `sar -r` コマンドを使用すると、現在使用されていないメモリーページ数とスワップファイルのディスクブロック数が表示されます。

```
$ sar -r
00:00:00 freemem freeswap
01:00:00    2135    401922
```

次の表では、`-r` オプションからの出力について説明します。

freemem	コマンドによるサンプル収集間隔の間にユーザープロセスに利用できる平均メモリーページ数。ページサイズはマシンに応じて異なる
freeswap	ページスワップに使用可能な 512 バイトのディスクブロック数

例 3-14 未使用のメモリーをチェックする (sar -r)

次の例は、`sar -r` コマンドからの出力を示します。

```
$ sar -r
SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:04 freemem freeswap
01:00:00    44717    1715062
02:00:01    44733    1715496
03:00:00    44715    1714746
04:00:00    44751    1715403
05:00:00    44784    1714743
06:00:00    44794    1715186
07:00:00    44793    1715159
```

```

08:00:00  44786  1714914
08:20:00  44805  1715576
08:40:01  44797  1715347
09:00:00  44761  1713948
09:20:00  44802  1715478
09:40:00  41770  1682239
10:00:00  35401  1610833
10:20:00  34295  1599141
10:40:00  33943  1598425
11:00:00  30500  1561959

Average   43312  1699242

```

▼ CPU の使用状況をチェックする方法 (sar -u)

- **sar -u** コマンドを使用すると、CPU 使用状況の統計が表示されます。

```

$ sar -u
00:00:00  %usr  %sys  %wio  %idle
01:00:00    0    0    0    100

```

オプションを指定しない **sar** コマンドは、**sar -u** コマンドと同じです。プロセッサの状態には、「ビジー」と「アイドル」があります。ビジー状態のときは、プロセッサはユーザーモードまたはシステムモードになっています。アイドル状態のときは、プロセッサは入出力の完了を待っているか、何も処理することがないので「待機」している状態です。

次の表では、**-u** オプションからの出力について説明します。

%usr	プロセッサがユーザーモードになっている時間の割合が表示されます。
%sys	プロセッサがシステムモードになっている時間の割合が表示されます。
%wio	プロセッサがアイドル状態で入出力の完了を待っている時間の割合が表示されます。
%idle	プロセッサがアイドル状態で入出力を待っていない時間の割合が表示されます。

一般に、**%wio** の値が大きい場合は、ディスクの処理速度が低下していることを意味します。

例 3-15 CPU の使用状況をチェックする (sar -u)

次の例は、**sar -u** コマンドからの出力を示します。

```

$ sar -u
00:00:04  %usr  %sys  %wio  %idle
01:00:00    0    0    0    100
02:00:01    0    0    0    100
03:00:00    0    0    0    100

```

```

04:00:00      0      0      0      100
05:00:00      0      0      0      100
06:00:00      0      0      0      100
07:00:00      0      0      0      100
08:00:00      0      0      0      100
08:20:00      0      0      0      99
08:40:01      0      0      0      99
09:00:00      0      0      0      99
09:20:00      0      0      0      99
09:40:00      4      1      0      95
10:00:00      4      2      0      94
10:20:00      1      1      0      98
10:40:00     18      3      0      79
11:00:00     25      3      0      72

Average        2      0      0      98

```

▼ システムテーブルのステータスをチェックする方法 (sar -v)

- **sar -v** コマンドを使用すると、プロセステーブル、iノードテーブル、ファイルテーブル、および共有メモリーレコードテーブルのステータスが表示されます。

```

$ sar -v
00:00:00  proc-sz   ov  inod-sz   ov  file-sz   ov  lock-sz
01:00:00  43/922    0 2984/4236  0  322/322   0   0/0

```

例 3-16 システムテーブルのステータスをチェックする (sar -v)

次の例は、sar -v コマンドからの一部省略した出力を示します。この例は、すべてのテーブルに十分なサイズがあり、オーバーフローは発生しないことを示します。これらのテーブルには、いずれも物理メモリーの容量に基づいて領域が動的に割り当てられます。

```

$ sar -v

00:00:04  proc-sz   ov  inod-sz   ov  file-sz   ov  lock-sz
01:00:00  69/8010   0 3476/34703  0   0/0       0   0/0
02:00:01  69/8010   0 3476/34703  0   0/0       0   0/0
03:00:00  69/8010   0 3476/34703  0   0/0       0   0/0
04:00:00  69/8010   0 3494/34703  0   0/0       0   0/0
05:00:00  69/8010   0 3494/34703  0   0/0       0   0/0
06:00:00  69/8010   0 3494/34703  0   0/0       0   0/0
07:00:00  69/8010   0 3494/34703  0   0/0       0   0/0
08:00:00  69/8010   0 3494/34703  0   0/0       0   0/0
08:20:00  69/8010   0 3494/34703  0   0/0       0   0/0
08:40:01  69/8010   0 3494/34703  0   0/0       0   0/0
09:00:00  69/8010   0 3494/34703  0   0/0       0   0/0
09:20:00  69/8010   0 3494/34703  0   0/0       0   0/0
09:40:00  74/8010   0 3494/34703  0   0/0       0   0/0
10:00:00  75/8010   0 4918/34703  0   0/0       0   0/0
10:20:00  72/8010   0 4918/34703  0   0/0       0   0/0
10:40:00  71/8010   0 5018/34703  0   0/0       0   0/0
11:00:00  77/8010   0 5018/34703  0   0/0       0   0/0

```

次の表に `-v` オプションからの出力について説明します。

フィールド名	説明
<code>proc-sz</code>	現在カーネル内で使用されているか割り当てられている、プロセスエントリ (<code>proc</code> 構造体) の数。
<code>inod-sz</code>	メモリー内の合計 <code>i</code> ノード数とカーネル内で割り当て済みの最大 <code>i</code> ノード数の比。これは厳密な上限ではありません。ここからオーバーフローすることもあります。
<code>file-sz</code>	開いているシステムファイルテーブルのサイズ。ファイルテーブルには領域が動的に割り当てられるので、 <code>sz</code> は <code>0</code> として表示されます。
<code>ov</code>	各テーブルのサンプルポイント間で発生しているオーバーフロー。
<code>lock-sz</code>	現在カーネル内で使用されているか割り当てられている、共有メモリーレコードテーブルのエントリ数。共有メモリーレコードテーブルには領域が動的に割り当てられるので、 <code>sz</code> は <code>0</code> として表示されます。

▼ スワップ動作をチェックする方法 (`sar -w`)

- `sar -w` コマンドを使用すると、スワッピングと切り替え動作が表示されます。

```
$ sar -w
00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00  0.00    0.0    0.00    0.0    22
```

次に、`sar -w` コマンド出力の対象となる値と説明を示します。

<code>swpin/s</code>	メモリーに転送される 1 秒当たりの軽量プロセス数
<code>bswin/s</code>	スワップイン用に転送される 1 秒当たりのブロック数。 /* (float)PGTOBLK(xx->cvmi.pgswpin) / sec_diff */.
<code>swpot/s</code>	メモリーからスワップアウトされる 1 秒当たりの平均プロセス数。この数値が 1 より大きい場合は、メモリーを増やす必要がある
<code>bswot/s</code>	スワップアウト用に転送される 1 秒当たりのブロック数
<code>pswch/s</code>	1 秒当たりのカーネルスレッド切り替え数

注- すべてのプロセスのスワップインには、プロセスの初期化が含まれます。

例 3-17 スワップ動作をチェックする (sar -w)

次の例は、sar -w コマンドからの出力を示します。

```
$ sar -w

00:00:04 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 132
02:00:01 0.00 0.0 0.00 0.0 133
03:00:00 0.00 0.0 0.00 0.0 133
04:00:00 0.00 0.0 0.00 0.0 134
05:00:00 0.00 0.0 0.00 0.0 133
06:00:00 0.00 0.0 0.00 0.0 133
07:00:00 0.00 0.0 0.00 0.0 132
08:00:00 0.00 0.0 0.00 0.0 131
08:20:00 0.00 0.0 0.00 0.0 133
08:40:01 0.00 0.0 0.00 0.0 132
09:00:00 0.00 0.0 0.00 0.0 132
09:20:00 0.00 0.0 0.00 0.0 132
09:40:00 0.00 0.0 0.00 0.0 335
10:00:00 0.00 0.0 0.00 0.0 601
10:20:00 0.00 0.0 0.00 0.0 353
10:40:00 0.00 0.0 0.00 0.0 747
11:00:00 0.00 0.0 0.00 0.0 804

Average 0.00 0.0 0.00 0.0 198
```

▼ 端末動作をチェックする方法 (sar -y)

- sar -y コマンドを使用すると、端末デバイスの動作を監視できます。

```
$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00 0 0 0 0 0 0
```

大量の端末入出力がある場合は、このレポートを使用して不良な回線がないかどうかを判別できます。次に、記録される動作を示します。

rawch/s 1秒当たりの入力文字数 (raw 待ち行列)。
 canch/s 標準待ち行列で処理される 1秒当たりの入力文字数。
 outch/s 1秒当たりの出力文字数 (出力待ち行列)。
 rcvin/s 1秒当たりの受信側ハードウェア割り込み数。
 xmtin/s 1秒当たりの送信側ハードウェア割り込み数。
 mdmin/s 1秒当たりのモデム割り込み数。

1秒当たりのモデム割り込み数 (mdmin/s) は、0に近い値になります。また、1秒当たりの送受信側ハードウェア割り込み数 (xmtin/s と rcvin/s) は、それぞれ着信または発信文字数以下になります。そうでない場合は、不良回線がないかどうかをチェックしてください。

例 3-18 端末動作をチェックする (sar -y)

次の例は、sar -y コマンドからの出力を示します。

```
$ sar -y

00:00:04 rawch/s  canch/s  outch/s  rcvin/s  xmtin/s  mdmin/s
01:00:00      0      0      0      0      0      0
02:00:01      0      0      0      0      0      0
03:00:00      0      0      0      0      0      0
04:00:00      0      0      0      0      0      0
05:00:00      0      0      0      0      0      0
06:00:00      0      0      0      0      0      0
07:00:00      0      0      0      0      0      0
08:00:00      0      0      0      0      0      0
08:20:00      0      0      0      0      0      0
08:40:01      0      0      0      0      0      0
09:00:00      0      0      0      0      0      0
09:20:00      0      0      0      0      0      0
09:40:00      0      0      1      0      0      0
10:00:00      0      0      37     0      0      0
10:20:00      0      0      0      0      0      0
10:40:00      0      0      3      0      0      0
11:00:00      0      0      3      0      0      0

Average      0      0      1      0      0      0
```

▼ システム全体のパフォーマンスをチェックする方法 (sar -A)

- **sar -A** コマンドを使用すると、すべてのオプションを指定した場合と同じように、システム全体のパフォーマンスを示す統計情報が表示されます。このコマンドを使用すると、全体像を把握できます。複数のタイムセグメントからのデータが表示される場合は、レポートに平均値が含まれます。

システム動作データの自動収集 (sar)

システム動作データを自動的に収集するには、3つのコマンドを使用します。sadc、sa1、およびsa2です。

sadc データ収集ユーティリティーは、システム動作に関するデータを定期的に収集し、24時間ごとに1つのファイルに2進形式で保存します。sadc コマンドを定期的に (通常は1時間ごとに) 実行したり、システムがマルチユーザーモードでブートするときにも実行するように設定できます。データファイルは、/var/adm/sa ディレクトリに格納されます。各ファイルには sadd という名前が与えられます。この場合、dd は現在の日付です。このコマンドの書式は次のとおりです。

```
/usr/lib/sa/sadc [t n] [ofile]
```

このコマンドは、 t 秒(5秒より長くする必要がある)間隔でサンプルデータを n 回収集します。このコマンドは次に、2進形式の *ofile* ファイルまたは標準出力に書き込みます。

ブート時の **sadc** コマンドの実行

カウンタが0にリセットされたときから統計情報を記録するために、**sadc** コマンドをシステムのブート時に実行する必要があります。**sadc** をブート時に確実に実行するために、**svcadm enable system/sar:default** コマンドで日次データファイルにレコードを書き込みます。

コマンドエントリの書式は次のとおりです。

```
/usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa'date +%d"
```

sa1 スクリプトを使用した **sadc** コマンドの定期的な実行

定期的にレコードを生成するには、**sadc** コマンドを定期的に実行する必要があります。そのためには、**/var/spool/cron/crontabs/sys** ファイルの次の行をコメント解除するのがもっとも簡単な方法です。

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

sys の **crontab** エントリによって、次のように動作します。

- 最初の2つの **crontab** エントリによって、月曜から金曜までの午前8時から午後5時までは20分ごとに、それ以外では1時間ごとに、レコードが **/var/adm/sa/sadd** ファイルに書き込まれます。
- 3番目のエントリは、月曜から金曜までは1時間ごとに、レコードを **/var/adm/sa/saradd** ファイルに書き込み、すべての **sar** オプションが含まれます。

これらのデフォルトは、必要に応じて変更できます。

sa2 スクリプトを使用したレポートの生成

もう1つのシェルスクリプト **sa2** は、2進データファイルでないレポートを生成します。**sa2** コマンドは **sar** コマンドを呼び出して、レポートファイルに ASCII 出力を書き込みます。

データの自動収集を設定する (**sar**)

sar コマンドを使用すると、システム動作データそのものを収集するか、**sadc** コマンドで作成された日次動作ファイルに収集された情報を報告できます。

sar コマンドの書式は次のとおりです。

```
sar [-aAbcdgkmpqruvw] [-o file] t [n]
```

```
sar [-aAbcdgkmpqruvw] [-s time] [-e time] [-i sec] [-f file]
```

次の sar コマンドは、オペレーティングシステム内の累積動作カウンタから t 秒間隔で n 回データを収集します。 t は、5 秒以上の値にします。それ以外の値にすると、コマンドそのものがサンプルに影響を与えることがあります。また、サンプルの収集間隔を指定する必要があります。指定しないと、このコマンドは第2の書式に従って動作します。 n のデフォルト値は1です。次の例では、10 秒間隔で2つのサンプルが収集されます。`-o` オプションを指定すると、サンプルは2進形式でファイルに保存されます。

```
$ sar -u 10 2
```

その他に、sar では次の点に注意する必要があります。

- サンプル間隔またはサンプル数を指定しなければ、sar コマンドは以前に記録されたファイルからデータを抽出します。このファイルは、`-f` オプションで指定したファイル、またはデフォルトでは最新日付分の標準日次動作ファイル `/var/adm/sa/sadd` です。
- `-s` オプションと `-e` オプションでは、レポートの開始時間と終了時間を定義します。開始時間と終了時間の書式は `hh[:mm[:ss]]` です (この場合、`hh`、`mm`、`ss` はそれぞれ時間、分、秒を表します)。
- `-i` オプションでは、レコードの選択間隔を秒単位で指定します。`-i` オプションを指定しなければ、日次動作ファイル内で見つかったすべての間隔がレポートされます。

次の表に、sar コマンドのオプションとその動作を示します。

表3-5 sar コマンドのオプション

オプション	アクション
-a	ファイルアクセス操作をチェックする
-b	バッファ動作をチェックする
-c	システムコールをチェックする
-d	各ブロックデバイスの動作をチェックする
-g	ページアウトとメモリの解放をチェックする
-k	カーネルメモリの割り当てをチェックする
-m	プロセス間通信をチェックする

表 3-5 sar コマンドのオプション (続き)

オプション	アクション
-nv	システムテーブルのステータスをチェックする
-p	スワップとディスパッチ動作をチェックする
-q	待ち行列動作をチェックする
-r	未使用メモリーをチェックする
-u	CPU の使用率をチェックする
-w	ボリュームのスワッピングと切り替えをチェックする
-y	端末動作をチェックする
-A	システム全体のパフォーマンスをレポートする (すべてのオプションを入力した場合と同じです)

オプションを使用しなければ、sar コマンドを -u オプションを指定して呼び出すのと同じです。

▼ 自動データ収集を設定する方法

1 root 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 svcadm enable system/sar:default コマンドを実行します。

このバージョンの sadc コマンドは、カウンタが 0 にリセットされる時間 (ブート時) を示す特殊なレコードを書き込みます。

3 crontab ファイル /var/spool/cron/crontabs/sys を編集します。

注 - crontab ファイルは直接編集しないでください。既存の crontab ファイルを変更するときには、代わりに crontab -e コマンドを使用してください。

```
# crontab -e sys
```

4 次の行のコメントを解除します。

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

詳細は、[crontab\(1\)](#) のマニュアルページを参照してください。

◆◆◆ 第 4 章

システムタスクのスケジュール設定 (タスク)

この章では、`crontab` コマンドおよび `at` コマンドを使用して、定型タスクや 1 度限りのシステムタスクをスケジュール設定する方法を説明します。

また、次のファイルを使用して上記のコマンドの使用を制御する方法も説明します。

- `cron.deny`
- `cron-allow`
- `at.deny`

この章の内容は次のとおりです。

- 83 ページの「システムタスクを自動的に実行する方法」
- 85 ページの「システムタスクのスケジュール設定」
- 96 ページの「`at` コマンドを使用したタスクのスケジューリング」

システムタスクを自動的に実行する方法

多数のシステムタスクを自動的に実行するよう設定できます。これらのタスクの中には、定期的な実行が必要になるタスクがあります。また、夜間や週末などの就業時間外に 1 回だけ実行するタスクもあります。

このセクションでは、`crontab` と `at` という 2 つのコマンドについて概説します。これらのコマンドを使用すると、定型タスクをスケジュールして、自動的に実行することができます。`crontab` コマンドは、繰り返し実行するコマンドをスケジュールします。`at` コマンドは、1 回だけ実行するタスクをスケジュールします。

次の表は、`crontab` と `at`、およびこれらのコマンドの使用を制御できるファイルをまとめたものです。

表 4-1 コマンドのサマリー: システムタスクのスケジューリング

コマンド	スケジューリングの対象	ファイルの格納場所	アクセスを制御するファイル
crontab	一定間隔で実行する複数のシステムタスク	/var/spool/cron/crontabs	/etc/cron.d/cron.allow および /etc/cron.d/cron.deny
at	1つのシステムタスク	/var/spool/cron/atjobs	/etc/cron.d/at.deny

繰り返されるジョブのスケジューリング (crontab)

定型的なシステム管理タスクは、`crontab` コマンドを使用して、毎日、毎週、または毎月それぞれ1回ずつ実行するようにスケジュールできます。

毎日1回の `crontab` システム管理タスクには、次のようなものがあります。

- 作成後、数日以上経過したファイルを一時ディレクトリから削除する
- アカウンティングサマリーコマンドを実行する
- `df` コマンドおよび `ps` コマンドを使用してシステムのスナップショットを取る
- 日常のセキュリティー監視を実行する
- システムのバックアップを実行する

毎週1回の `crontab` システム管理タスクには、次のようなものがあります。

- `man -k` コマンドで処理する `catman` データベースを再構築する
- `fsck -n` コマンドを実行して、ディスク問題があれば表示する

毎月1回の `crontab` システム管理タスクには、次のようなものがあります。

- 指定月に使用されなかったファイルをリストする
- 月次アカウンティングレポートを生成する

上記に加えて、連絡事項の通知やバックアップファイルの削除などの定型的システムタスクを実行するように、`crontab` コマンドをスケジュールすることもできます。

`crontab` ジョブをスケジュールする手順については、89 ページの「[crontab ファイルを作成または編集する方法](#)」を参照してください。

1つのジョブのスケジューリング (at)

`at` コマンドを使用すると、1つのジョブをあとで実行するようにスケジュールできます。ジョブは1つのコマンドやスクリプトで構成されます。

crontabと同様に、at コマンドを使用すると定型タスクの自動実行をスケジュールできます。しかし、crontab ファイルとは異なり、at ファイルはタスクを1回だけ実行します。その後はディレクトリから削除されます。したがって、at コマンドが役立つのは、単純なコマンドまたはスクリプトを実行して、別ファイルに書き出した出力をあとから調べるような場合です。

at ジョブの実行を指定するには、コマンドを入力してから、at コマンド構文に従ってオプションで実行時間を指定してください。at ジョブの実行方法については、97 ページの「[at コマンドの説明](#)」を参照してください。

at コマンドは、入力されたコマンドまたはスクリプトを、現在の環境変数のコピーと一緒に /var/spool/cron/atjobs ディレクトリに格納します。作成された at ジョブには、ファイル名として、at 待ち行列内での位置を指定する長い数値と .a 拡張子からなる、たとえば 793962000.a のような文字列が与えられます。

cron デーモンは、起動時に at ジョブをチェックし、新しく実行されるジョブを待機します。cron デーモンが at ジョブを実行すると、atjobs ディレクトリから at ジョブのファイルが削除されます。詳細は、[at\(1\)](#) のマニュアルページを参照してください。

at ジョブをスケジュールする手順については、98 ページの「[at ジョブを作成する方法](#)」を参照してください。

システムタスクのスケジュール設定

このセクションでは、crontab ファイルを使用してシステムタスクをスケジュールするタスクについて説明します。

crontab ファイルの作成と編集 (タスクマップ)

タスク	説明	参照先
crontab ファイルを作成または編集する	crontab -e コマンドを使用し、crontab ファイルを作成または編集する	89 ページの「 crontab ファイルを作成または編集する方法 」
crontab ファイルが存在するかどうかを確認する	ls -l コマンドを使用し、/var/spool/cron/crontabs ファイルの内容を確認する	90 ページの「 crontab ファイルを確認する方法 」
crontab ファイルを表示する	ls -l コマンドを使用し、crontab ファイルを表示する	91 ページの「 crontab ファイルを表示する方法 」

タスク	説明	参照先
crontab ファイルを削除する	crontab ファイルには、アクセス制限がかけられている。したがって、crontab ファイルを削除するには、rm ではなく crontab -r コマンドを使用する	92 ページの「 crontab ファイルを削除する方法 」
crontab の使用を拒否する	ユーザーによる crontab コマンドの使用を拒否するには、/etc/cron.d/cron.deny ファイルを編集してユーザー名を追加する	94 ページの「 crontab コマンドの使用を拒否する方法 」
crontab の使用を特定のユーザーに限定する	crontab コマンドのユーザーの使用を許可するには、/etc/cron.d/cron.allow ファイルにユーザー名を追加する	94 ページの「 crontab コマンドの使用を特定のユーザーに限定する方法 」

繰り返されるシステムタスクのスケジューリング (cron)

以降のセクションで、crontab ファイルをどのように作成、編集、表示、削除するか、さらに、それらのファイルの使用をどのように制御するかを説明します。

crontab ファイルの内容

cron デーモンは、各 crontab ファイル内にあるコマンドに従ってシステムタスクをスケジュールします。crontab ファイルには、それぞれ一定間隔で実行されるコマンドが 1 行に 1 つずつ入っています。各行の先頭は cron デーモンが各コマンドを実行する日時情報です。

たとえば、SunOS ソフトウェアのインストール時に root という名前の crontab ファイルが提供されます。このファイルの内容には、次のコマンド行が含まれています。

```
10 3 * * * /usr/sbin/logadm      (1)
15 3 * * 0 /usr/lib/fs/nfs/nfsfind  (2)
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1      (3)
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean  (4)
```

これらのコマンド行の出力について次に説明します。

- 最初の行は、毎日午前 3 時 10 分に logadm コマンドを実行します。
- 2 行目は、毎週日曜日の午前 3 時 15 分に nfsfind スクリプトを実行します。
- 3 行目は、毎日午前 2 時 10 分に、夏時間をチェック (して必要に応じて修正) するスクリプトを実行します。

RTCタイムゾーンも `/etc/rtc_config` ファイルもない場合、このエントリは何もしません。

x86のみ `-usr/sbin/rtc` スクリプトは、x86 ベースのシステムでのみ実行できます。

- 4行目は、毎日午前3時30分に Generic Security Service テーブル `/etc/gss/gsscred_db` の重複エントリをチェック (重複エントリがある場合は削除) します。

`crontab` ファイル内のコマンド行の構文の詳細は、[88 ページの「`crontab` ファイルエントリの構文](#)」を参照してください。

`crontab` ファイルは `/var/spool/cron/crontabs` ディレクトリに保存されます。SunOS ソフトウェアのインストール時には、`root` 以外にもいくつかの `crontab` ファイルが提供されます。次の表を参照してください。

表 4-2 デフォルトの `crontab` ファイル

<code>crontab</code> ファイル	機能
<code>adm</code>	アカウントिंग
<code>root</code>	一般的なシステム機能とファイルシステムの整理
<code>sys</code>	パフォーマンスデータの収集
<code>uucp</code>	一般的な <code>uucp</code> の整理

デフォルトの `crontab` ファイルの他に、ユーザーは `crontab` ファイルを作成してユーザー自身のシステムタスクをスケジュールできます。その他の `crontab` ファイルは、作成したユーザーのアカウントに基づいて、`bob`、`mary`、`smith`、`jones` などのように命名されます。

`root` またはほかのユーザーの `crontab` ファイルを使用するには、スーパーユーザーの特権が必要です。

`crontab` ファイルの作成、編集、表示、削除の手順については、以降のセクションで説明します。

cron デーモンのスケジューリング管理

`cron` デーモンは、`crontab` コマンドの自動スケジューリングを管理します。`cron` デーモンは、`/var/spool/cron/crontab` ディレクトリに `crontab` ファイルがあるかどうかをチェックします。

cron デーモンは、起動時に次のタスクを実行します。

- 新しい crontab ファイルがないかを確認する
- ファイル内のリストから実行時間を読み取る
- 正しい時間にコマンドを実行する
- 更新された crontab ファイルに関する crontab コマンドからの通知を待機する

ほとんど同様に、cron デーモンは at ファイルのスケジューリングを制御します。これらのファイルは /var/spool/cron/atjobs ディレクトリに格納されています。cron デーモンは、実行された at ジョブに関する crontab コマンドからの通知も待機します。

crontab ファイルエントリの構文

crontab ファイルは、1 行に 1 つのコマンドが入っており、各コマンド行の最初の 5 つのフィールドでは、コマンドが実行される時間を指定します。これら 5 つのフィールドを、次の表で説明します。これらのフィールドはスペース (空白) で区切られます。

表 4-3 crontab 時間フィールドの値

時間フィールド	値
分	0-59
時	0-23
日	1-31
月	1-12
曜日	0 - 6 (0 は日曜日)

次に、crontab 時間フィールドで特殊文字を使用する際のガイドラインを示します。

- 各フィールドはスペースで区切る
- 複数の値の間はコンマで区切る
- 値の範囲はハイフンを使用して指定する
- 取り得るすべての値を含むには、ワイルドカードとしてアスタリスクを使用する
- コメントまたは空白行を示すには、行の先頭にコメント記号 (#) を使用する

たとえば、次の crontab コマンドエントリは、毎月 1 日と 15 日の午後 4 時に、ユーザーのコンソールウィンドウに注意を促すメッセージを表示します。

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

crontab ファイル内の各コマンドは、長くても 1 行内に入れる必要があります。crontab ファイルは余分なキャリッジリターンを認識しません。crontab のエントリとコマンドオプションの詳細は、[crontab\(1\)](#) のマニュアルページを参照してください。

crontab ファイルの作成と編集

crontab ファイルを作成するもっとも簡単な方法は、`crontab -e` コマンドを使用することです。このコマンドは、システム環境で指定されたテキストエディタを起動します。システム環境のデフォルトのエディタは、`EDITOR` 環境変数で定義されます。この環境変数が設定されていない場合は、`crontab` コマンドはデフォルトのエディタ `ed` を使用します。あらかじめ、使い慣れたエディタを選択しておく必要があります。

次の例は、エディタが定義されているかどうかを確認する方法と、`vi` をデフォルトのエディタとして設定する方法を示しています。

```
$ which $EDITOR
$
$ EDITOR=vi
$ export EDITOR
```

`crontab` ファイルを作成すると、自動的に `/var/spool/cron/crontabs` ディレクトリ内に格納され、作成者のユーザー名で命名されます。`root` 特権があれば、ほかのユーザーや `root` の `crontab` ファイルを作成または編集できます。

▼ crontab ファイルを作成または編集する方法

始める前に

`root` またはほかのユーザーに属する `crontab` ファイルを作成または編集する場合は、`root` 役割になる必要があります。『[Oracle Solaris 11.1 の管理: セキュリティサービス](#)』の「[割り当てられている管理権限を使用する方法](#)」を参照してください。

自分の `crontab` ファイルを編集する場合は、`root` 役割になる必要はありません。

- 1 新しい `crontab` ファイルを作成するか、既存の `crontab` ファイルを編集します。

```
# crontab -e [username]
```

`username` は、`crontab` ファイルを作成または編集するユーザーのアカウント名を指定します。自分の `crontab` ファイルを作成するにはスーパーユーザー特権は必要ありませんが、`root` または別のユーザーの `crontab` ファイルを作成したり編集したりするには、スーパーユーザー特権が必要です。



注意 - 誤ってオプションを指定しないで `crontab` コマンドを入力した場合は、使用しているエディタの中断文字を入力してください。この文字を入力すると、変更結果を保存せずに `crontab` コマンドを終了できます。この場合に変更結果を保存してファイルを終了すると、既存の `crontab` ファイルが空のファイルで上書きされます。

- 2 コマンド行を **crontab** ファイルに追加します。

88 ページの「**crontab** ファイルエントリの構文」に記載されている構文に従ってください。crontab ファイルは、`/var/spool/cron/crontabs` ディレクトリに保存されます。

- 3 **crontab** ファイルの変更箇所を確認します。

```
# crontab -l [username]
```

例 4-1 crontab ファイルを作成する

次の例は、他のユーザーのための crontab ファイルをどのように作成するかを示します。

```
# crontab -e jones
```

次のコマンドエントリを新しい crontab ファイルに追加すると、毎週日曜日の午前 1 時にユーザーのホームディレクトリからすべてのログファイルが自動的に削除されます。このコマンドエントリは出力先を変更しないので、出力先変更文字がコマンド行の `*.log` のあとに追加されます。このためコマンドが正しく実行されます。

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/jones/*.log > /dev/null 2>&1
```

▼ crontab ファイルを確認する方法

- 特定のユーザーの crontab ファイルがあるかどうかを確認するには、`/var/spool/cron/crontabs` ディレクトリで `ls -l` コマンドを使用します。たとえば、次の出力はユーザー `jones` と `smith` の crontab ファイルがあることを示しています。

```
$ ls -l /var/spool/cron/crontabs
```

ユーザーの crontab ファイルの内容を確認するには、`crontab -l` コマンドを使用します。91 ページの「**crontab** ファイルを表示する方法」を参照してください。

crontab ファイルの表示

`crontab -l` コマンドは、`cat` コマンドが他のファイルタイプの内容を表示するのと同様に、crontab ファイルの内容を表示します。このコマンドを使用するために、(crontab ファイルが入っている)`/var/spool/cron/crontabs` ディレクトリに移動する必要はありません。

デフォルトでは、`crontab -l` コマンドは自分自身の crontab ファイルを表示します。他のユーザーの crontab ファイルは、スーパーユーザーでなければ表示できません。

▼ crontab ファイルを表示する方法

始める前に root またはほかのユーザーに属する crontab ファイルを表示するには、root 役割になります。『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

自分の crontab ファイルを表示する場合は、root 役割になる必要はありません。

- 次のように入力して、crontab ファイルを表示します。

```
# crontab -l [username]
```

username は、crontab ファイルを表示するユーザーのアカウント名を指定します。他のユーザーの crontab ファイルを表示するには、スーパーユーザー特権が必要です。



注意 - 誤ってオプションを指定しないで crontab コマンドを入力した場合は、使用しているエディタの中断文字を入力してください。この文字を入力すると、変更結果を保存せずに crontab コマンドを終了できます。この場合に変更結果を保存してファイルを終了すると、既存の crontab ファイルが空のファイルで上書きされます。

例 4-2 crontab ファイルを表示する

次の例は、crontab -l コマンドを使用してユーザーのデフォルトの crontab ファイルを表示する方法を示します。

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
```

例 4-3 デフォルトの root の crontab ファイルを表示する

次の例は、デフォルトの root の crontab ファイルを表示する方法を示します。

```
$ suPassword:

# crontab -l
#ident "@(#)root 1.19 98/07/06 SMI" /* SVr4.0 1.1.3.1 */
#
# The root crontab should be used to perform accounting data collection.
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

例 4-4 他のユーザーの crontab ファイルを表示する

次の例は、他のユーザーの crontab ファイルを表示する方法を示します。

```
$ su
Password:
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null 2>&1
```

crontab ファイルの削除

デフォルトでは、rm コマンドを使用して誤って crontab ファイルを削除してしまうことがないように、crontab ファイルは保護されています。crontab ファイルを削除する場合は、rm コマンドではなく crontab -r コマンドを使用してください。

デフォルトでは、crontab -r コマンドは自分自身の crontab ファイルを削除します。

このコマンドを使用するために、/var/spool/cron/crontabs ディレクトリ (crontab ファイルが入っている) に移動する必要はありません。

▼ crontab ファイルを削除する方法

始める前に root またはほかのユーザーに属する crontab ファイルを削除するには、root 役割になります。役割には、認証と特権コマンドが含まれます。『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

自分の crontab ファイルを削除する場合は、root 役割になる必要はありません。

- 1 次のように入力して、crontab ファイルを削除します。

```
# crontab -r [username]
```

username は、crontab ファイルを削除するユーザーのアカウント名を指定します。他のユーザーの crontab ファイルを削除するには、スーパーユーザーの権限が必要です。



注意-誤ってオプションを指定しないで crontab コマンドを入力した場合は、使用しているエディタの中断文字を入力してください。この文字を入力すると、変更結果を保存せずに crontab コマンドを終了できます。この場合に変更結果を保存してファイルを終了すると、既存の crontab ファイルが空のファイルで上書きされます。

- 2 crontab ファイルが削除されたことを確認します。

```
# ls /var/spool/cron/crontabs
```

例 4-5 crontab ファイルを削除する

次の例では、ユーザー smith が crontab -r コマンドを使用して自分の crontab ファイルを削除します。

```
$ ls /var/spool/cron/crontabs
adm   jones   root    smith   sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm   jones   root    sys     uucp
```

crontab コマンドの使用制御

/etc/cron.d ディレクトリ内の2つのファイルを使用して、crontab コマンドの使用を制御できます。これらのファイルは cron.deny および cron.allow です。これらのファイルによって、指定したユーザーだけが、それぞれ自分の crontab ファイルの作成、編集、表示、または削除などの crontab コマンドのタスクを実行できるようにします。

cron.deny ファイルおよび cron.allow ファイルは、それぞれ1行に1ユーザー名が入ったリストからなります。

これらの使用制御用ファイルは、次のように連携して機能を果たします。

- cron.allow が存在する場合は、このファイルにリストされているユーザーだけが crontab ファイルを作成、編集、表示、または削除できます。
- cron.allow が存在しない場合は、cron.deny にリストされているユーザーを除くすべてのユーザーが crontab ファイルを使用できます。
- cron.allow も cron.deny も存在しない場合は、crontab コマンドの実行にスーパーユーザーの権限が必要です。

cron.deny と cron.allow ファイルを編集または作成するには、スーパーユーザーの権限が必要です。

SunOS ソフトウェアのインストール時に作成される cron.deny ファイルには、次のユーザー名が含まれます。

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

デフォルトの cron.deny ファイル内のユーザー名は、いずれも crontab コマンドを使用できません。このファイルを編集して、crontab コマンドの使用を拒否したいユーザー名を追加できます。

cron.allow のデフォルトファイルは提供されません。つまり、Oracle Solaris ソフトウェアをインストールした直後は、すべてのユーザー(デフォルトの cron.deny

ファイルにリストされているユーザーを除く)が `crontab` コマンドを使用できません。 `cron.allow` ファイルを作成した場合、そのユーザーだけが `crontab` コマンドを使用できます。

▼ `crontab` コマンドの使用を拒否する方法

- 1 `root` 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

```
$ su -  
Password:  
#
```

- 2 `/etc/cron.d/cron.deny` ファイルを編集し、ユーザー名を 1 行に 1 つずつ追加します。 `crontab` コマンドの使用を拒否するユーザーを入れます。

```
daemon  
bin  
smtp  
nuucp  
listen  
nobody  
noaccess  
username1  
username2  
username3  
.  
.  
.
```

- 3 `/etc/cron.d/cron.deny` ファイルに新しいエントリが含まれているか確認します。

```
# cat /etc/cron.d/cron.deny  
daemon  
bin  
nuucp  
listen  
nobody  
noaccess
```

▼ `crontab` コマンドの使用を特定のユーザーに限定する方法

- 1 `root` 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 `/etc/cron.d/cron.allow` ファイルを作成します。

- 3 **root** 役割を **cron.allow** ファイルに追加します。
このファイルに **root** を追加しないと、スーパーユーザーの **crontab** コマンドの使用が拒否されます。
- 4 ユーザー名を1行に1つずつ追加します。
crontab コマンドの使用を許可するユーザーを入れます。

```
root
username1
username2
username3
.
.
.
```

例 4-6 **crontab** コマンドの使用を特定のユーザーに限定する

次は、ユーザー **jones**、**temp**、および **visitor** に **crontab** コマンドを使用させない **cron.deny** ファイルの例です。

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

次は、**cron.allow** ファイルの例です。ユーザー **root**、**jones**、および **smith** だけが、**crontab** コマンドを使用できます。

```
$ cat /etc/cron.d/cron.allow
root
jones
smith
```

crontab コマンドの使用制限を確認する方法

特定のユーザーが **crontab** コマンドを使用できるかどうかを確認するには、そのユーザーのアカウントでログインして **crontab -l** コマンドを使用します。

```
$ crontab -l
```

そのユーザーが **crontab** を使用できて、すでに **crontab** ファイルを作成している場合は、そのファイルが表示されます。そのユーザーが **crontab** コマンドを使用できるが、**crontab** ファイルがない場合は、次のようなメッセージが表示されます。

```
crontab: can't open your crontab file
```

このユーザーは、cron.allow ファイル (が存在する場合) に含まれているか、cron.deny ファイルに含まれていません。

ユーザーが crontab コマンドを使用できない場合は、上記の crontab ファイルの有無にかかわらず、次のメッセージが表示されます。

```
crontab: you are not authorized to use cron. Sorry.
```

このメッセージは、ユーザーが cron.allow ファイル (が存在する場合) に含まれていないか、cron.deny ファイルに含まれていることを意味します。

at コマンドを使用したタスクのスケジューリング

このセクションでは、at コマンドを使用してシステムタスクをスケジュールするタスクについて説明します。

at コマンドの使用 (タスクマップ)

タスク	説明	参照先
at ジョブを作成する	次のような目的には、at コマンドを使用する。 <ul style="list-style-type: none"> ■ コマンド行から at ユーティリティを起動する ■ 実行したいコマンドまたはスクリプトを、1行に1つずつ入力する ■ at ユーティリティを終了し、ジョブを保存する 	98 ページの「at ジョブを作成する方法」
at 待ち行列を表示する	atq コマンドを使用し、at 待ち行列を表示する	99 ページの「at 待ち行列を表示する方法」
at ジョブを確認する	atq コマンドを使用し、特定のユーザーの at ジョブが待ち行列に入れられたかどうかを確認する	99 ページの「at ジョブを確認する方法」
at ジョブを表示する	at -l [job-id] コマンドを使用し、待ち行列に入れられた at ジョブを表示する	100 ページの「at ジョブを表示する方法」

タスク	説明	参照先
at ジョブを削除する	at -r [job-id] コマンドを使用し、待ち行列から at ジョブを削除する	100 ページの「at ジョブを削除する方法」
at コマンドの使用を拒否する	at コマンドの使用を拒否するには、/etc/cron.d/at.deny ファイルを編集する	101 ページの「at コマンドの使用を拒否する方法」

1つのシステムタスクのスケジューリング (at)

以降のセクションでは、at コマンドを使用して以下のタスクを実行する方法を説明します。

- 特定の時間にジョブ(コマンドとスクリプト)の実行をスケジュールする
- これらのジョブを表示および削除する
- at コマンドの使用を制御する

デフォルトでは、ユーザーはそれぞれ自分の at ジョブファイルを作成、表示、または削除できます。root または他のユーザーの at ファイルにアクセスするには、スーパーユーザーの権限が必要です。

at ジョブの実行を設定すると、ジョブ識別番号と .a 拡張子が与えられます。これがジョブのファイル名および待ち行列番号になります。

at コマンドの説明

at ジョブファイルを書き込むには、以下の手順を実行します。

1. コマンド実行時間を指定して at ユーティリティを起動します。
2. あとで実行させるコマンドまたはスクリプトを入力します。

注-このコマンドまたはスクリプトからの出力が重要な場合は、後で調べることができるように、出力内容を必ずファイルに書き込むようにしてください。

たとえば、次の at ジョブは、7月31日の真夜中に smith ユーザーアカウントからコアファイルを削除します。

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
commands will be executed using /bin/csh
job 933486300.a at Tue Jul 31 23:45:00 2004
```

at コマンドの使用制御

特定のユーザーだけが自分の at ジョブに関する待ち行列情報を作成、削除、または表示できるように、at コマンドの使用を制御するファイルを設定できます。at コマンドの使用を制御するファイルは /etc/cron.d/at.deny です。ここにはユーザー名が列挙 (1 行に 1 人) されています。このファイルに列挙されているユーザーは、at コマンドを使用できません。

Solaris ソフトウェアのインストール時に作成される at.deny ファイルには、次のユーザー名が含まれます。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

スーパーユーザーの特権があれば、at.deny ファイルを編集して、at の使用を制限したい他のユーザー名を追加できます。

▼ at ジョブを作成する方法

- 1 at ユーティリティを起動して、ジョブを実行したい時間を指定します。

```
$ at [-m] time [date]
```

-m ジョブ終了後にメールを送る

time ジョブをスケジュールしたい時刻を指定する。24 時間制を使用しない場合は、am または pm を追加する。使用できるキーワードは、midnight、noon、now。分単位の値の指定はオプション

date 月または曜日の英語名の最初の 3 文字以上、またはキーワード today または tomorrow を指定する

- 2 at プロンプトに、実行したいコマンドまたはスクリプトを 1 行に 1 つずつ入力します。
各行の終わりで Return キーを押すことにより、複数のコマンドを入力できます。
- 3 at ユーティリティを終了し、Control-D キーを押して at ジョブを保存します。
作成できた at ジョブは待ち行列番号を割り当てられ、それがそのジョブのファイル名にもなります。この番号は at ユーティリティの終了時に表示されます。

例 4-7 at ジョブを作成する

次の例は、ユーザー `jones` が自分のバックアップファイルを 7:30 pm に削除するように作成した `at` ジョブを示しています。彼女は、ジョブの終了後にメールメッセージを受け取れるように `-m` オプションを使用しています。

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-D
job 897355800.a at Thu Jul 12 19:30:00 2004
```

彼女は次のメールメッセージを受け取りました。このメッセージは `at` ジョブが終了したことを確認しています。

```
Your "at" job "rm /home/jones/*.backup"
completed.
```

次の例は、`jones` が土曜の午前 4 時に大規模な `at` ジョブをスケジュールする方法を示しています。ジョブの出力先は `big.file` という名前のファイルです。

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

▼ at 待ち行列を表示する方法

- `at` 待ち行列で実行を待っているジョブを確認するには、次に示すように `atq` コマンドを使用します。

```
$ atq
```

このコマンドは、その使用者が作成した `at` ジョブに関するステータス情報を表示します。

▼ at ジョブを確認する方法

- `at` ジョブが作成できたかどうかを確認するには、`atq` コマンドを使用します。次の例の `atq` コマンドは、`jones` の `at` ジョブが待ち行列に入っていることを確認しています。

```
$ atq
Rank    Execution Date    Owner    Job                Queue    Job Name
1st     Jul 12, 2004 19:30    jones    897355800.a        a        stdin
2nd     Jul 14, 2004 23:45    jones    897543900.a        a        stdin
3rd     Jul 17, 2004 04:00    jones    897732000.a        a        stdin
```

▼ at ジョブを表示する方法

- 自分の at ジョブの実行時間に関する情報を表示するには、**at -l** コマンドを使用します。

```
$ at -l [job-id]
```

-l *job-id* オプションで、ステータスを表示したいジョブの識別番号を指定します。

例 4-8 at ジョブを表示する

次の例は、at -l コマンドからの出力を示しています。このコマンドは、特定のユーザーが実行を依頼したすべてのジョブのステータスに関する情報を表示します。

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2004
897355800.a    Thu Jul 12 19:30:00 2004
897732000.a    Tue Jul 17 04:00:00 2004
```

次の例は、at -l コマンドに1つのジョブを指定して表示された出力を示しています。

```
$ at -l 897732000.a
897732000.a    Tue Jul 17 04:00:00 2004
```

▼ at ジョブを削除する方法

始める前に root またはほかのユーザーに属する at ジョブを削除するには、root 役割になります。役割には、認証と特権コマンドが含まれます。『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

自分の at ジョブを削除する場合は、root 役割になる必要はありません。

- 1 次のように入力して、at ジョブが実行される前に待ち行列から削除します。

```
# at -r [job-id]
```

-r *job-id* オプションで、削除したいジョブの識別番号を指定します。

- 2 **at -l** (または **atq**) コマンドを使用して、at ジョブが削除されていることを確認します。

at -l コマンドは、at 待ち行列に残っているジョブを表示します。識別番号を指定したジョブは、このリストに表示されないはずですが。

```
$ at -l [job-id]
```

例 4-9 at ジョブを削除する

次の例では、ユーザーが7月17日の午前4時に実行されるようにスケジュールした at ジョブを削除しようとしています。まず、このユーザーは at 待ち行列を表示してそのジョブの識別番号を探します。次に、そのジョブを at 待ち行列から削除します。最後に、at 待ち行列をもう一度表示して上記のジョブが削除されていることを確認します。

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2003
897355800.a    Thu Jul 12 19:30:00 2003
897732000.a    Tue Jul 17 04:00:00 2003
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

▼ at コマンドの使用を拒否する方法

1 root 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 /etc/cron.d/at.deny ファイルを編集して、at コマンドを使用させないようにするユーザー名を1行に1つずつ追加します。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

例 4-10 at の使用を拒否する

次の例は、ユーザー smith と jones が at コマンドを使用できないように編集された at.deny ファイルです。

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
```

```
nobody
noaccess
jones
smith
```

▼ at コマンドの使用の拒否を確認する方法

- 特定のユーザー名が正しく `/etc/cron.d/at.deny` ファイルに追加されているかどうかを確認するには、そのユーザー名でログインして、`at -l` コマンドを使用します。ユーザー `smith` が `at` コマンドにアクセスできない場合、次のメッセージが表示されます。

```
# su smith
Password:
# at -l
at: you are not authorized to use at. Sorry.
```

同様に、そのユーザーが `at` ジョブの実行を依頼しようとした場合は、次のメッセージが表示されます。

```
# at 2:30pm
at: you are not authorized to use at. Sorry.
```

このメッセージによって、そのユーザーが `at.deny` ファイルに含まれていることが確認されます。

`at` コマンドを使用できる場合、`at -l` コマンドは何も返しません。

システムコンソール、端末デバイス、および電源サービスの管理 (タスク)

この章では、`ttymon` プログラムとシステム電源サービスを使用してシステムコンソールとローカル接続された端末デバイスを管理する方法について説明します。

この章の内容は次のとおりです。

- 103 ページの「システムコンソール、端末デバイス、および電源サービスの管理に関する新機能」
- 104 ページの「システムコンソールとローカル接続された端末デバイスの管理」
- 107 ページの「システム電源サービスの管理」

システムコンソール、端末デバイス、および電源サービスの管理に関する新機能

Oracle Solaris 11 では、次の機能が新しく導入または変更されています。

システム電源サービスの管理方法の変更

電源管理構成が SMF 構成リポジトリに移動しました。電源関連のコマンド、デーモン、および構成ファイルを組み合わせる代わりに、`poweradm` コマンドを使用してシステムの電源管理プロパティを直接管理します。これらの変更は、Oracle Solaris 11 で電源管理フレームワークを刷新するために広範囲で行われた変更の一部です。詳細は、107 ページの「システム電源サービスの管理」を参照してください。

システムコンソールとローカル接続された端末デバイスの管理

システムコンソールは、特別な属性を持ち、特定の目的に使用される端末です。たとえば、管理者向けのカーネルメッセージはコンソールに送信され、ほかの端末には送信されません。

端末は、Oracle Solaris とやりとりするための手段です。システムのビットマップグラフィックスディスプレイは、英数字端末と同じではありません。英数字端末はシリアルポートに接続され、テキストのみを表示します。グラフィックスディスプレイは、特別な手順に従って管理する必要はありません。

端末をコンピュータの物理的なモニターとキー配列に関連付けることもできます。グラフィックス端末がほかと異なる点は、コンピュータのグラフィックスカードとモニターに関連付ける必要があることです。したがって、シリアルポートに文字が送信されるのではなく、コンピュータに内蔵されているグラフィックスカードのメモリー上に文字が描画されます。

システムコンソールとローカル接続された端末デバイスを管理する SMF サービス

システムコンソールとローカル接続された端末デバイスは、SMF サービスのインスタンス (`svc:/system/console`) として表されます。このサービスはほとんどの動作を定義し、各インスタンスはサービスから継承された設定に対する優先指定値を持っています。ttymon プログラムは、これらの端末にログインサービスを提供するために使用されます。各端末では、ttymon プログラムの個別のインスタンスを使用します。サービスから ttymon プログラムに渡されるコマンド行引数によってプログラムの動作が制御されます。

システムに付属するサービスインスタンスは次のとおりです。

- `svc:/system/console-login:default`

デフォルトインスタンスは、常に ttymon プログラムによってシステムのハードウェアへのログインが提供されることを表します。具体例については、[105 ページの「システムコンソールの設定を変更する方法」](#)を参照してください。

- `svc:/system/console-login:{vt2, vt3, vt4, vt5, vt6 }`

システムの仮想コンソールには、追加のサービスインスタンスが提供されています。仮想コンソールが使用できない場合は、これらのサービスが自動的に無効化されます。詳細は、[vtdaemon\(1M\)](#) のマニュアルページを参照してください。

- `svc:/system/console-login:{terma, termb}`

`svc:/system/console-login:terma` および `svc:/system/console-login:termb` サービスは、便宜を図るために提供されます。これらのサービスは、追加の `/dev/term/a` および `/dev/term/b` ポートに対してログインサービスを設定するのに便利です。これらのサービスは、デフォルトで無効になっています。

追加のサービスインスタンスは、`svc:system/console-login` サービスの一部として定義できます。たとえば、`/dev/term/f` デバイスがあり、これをサポートする必要がある場合は、`svc:/system/console-login:termf` をインスタンス化して適切に構成することができます。

▼ システムコンソールの設定を変更する方法

システムコンソールの管理は SMF によって管理されます。システムコンソールのプロパティを設定するには、`svccfg` コマンドを使用します。

次の手順では、`svccfg` コマンドを使用してコンソールの端末タイプを変更する方法を説明します。

1 root 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

```
$ su -
Password:
#
```

2 svccfg コマンドを使用して、変更するサービスインスタンスのプロパティを設定します。

たとえば、`:default` サービスによって表されるシステムコンソールの端末タイプを変更するには、次のコマンドを入力します。

```
# svccfg -s svc:/system/console-login:default "setprop ttymon/terminal_type = xterm"
```



注意 - `svc:/system/console-login` の端末タイプを設定することは、すべてのインスタンスに変更の影響が及ぶため、お勧めできません。

▼ 補助端末に対してログインサービスを設定する方法

システム上の `/dev/term/a` または `/dev/term/b` シリアルポートに接続された端末に対しては、定義済みのサービスが提供されます。

`/dev/term/a` に対するログインサービスを有効にするには、次の手順に従います。

1 root 役割になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 次のようにしてサービスインスタンスを有効にします。

```
# svcadm enable svc:/system/console-login:terma
```

- 3 サービスがオンラインであることを確認します。

```
# svcs svc:/system/console-login:terma
```

サービスがオンラインであることが出力に表示されるはずですが、サービスが保守モードである場合は、サービスのログファイルで詳細を調べます。

▼ システム端末のボーレート速度を設定する方法

この手順では、コンソールのボーレート速度を設定する方法を示します。x86 ベースのシステムでのコンソール速度のサポートは、特定のプラットフォームに依存します。

SPARC ベースのシステムには、次のコンソール速度がサポートされています。

- 9600 bps
- 19200 bps
- 38400 bps

- 1 管理者になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 システムタイプに適したボーレート速度を設定するには、**eeeprom** コマンドを使用します。

```
# eeeprom ttya-mode=baud-rate,8,n,1,-
```

たとえば、x86 ベースシステムのコンソールのボーレート速度を 38400 に変更するには、次のように入力します。

```
# eeeprom ttya-mode=38400,8,n,1,-
```

- 3 **/etc/ttydefs** ファイルのコンソール行を次のように変更します。

```
console baud-rate hupcl opost onlcr:baud-rate::console
```

- 4 システムのタイプに合わせて、次の追加変更を行います。

これらの変更はプラットフォームに依存することに注意してください。

- **SPARC** ベースのシステム: **/etc/driver/drv** ディレクトリにあるバージョンの **options.conf** ファイルでボーレート速度を変更します。

ボーレートを 9600 に変更するには、次のコマンドを使用します。

```
# 9600 :bd:
ttymodes="2502:1805:bd:8a3b:3:1c:7f:15:4:0:0:11:13:1a:19:12:f:17:16";
```

ボーレート速度を 19200 に変更するには、次のコマンドを使用します。

```
# 19200          :be:
ttymodes="2502:1805:be:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

ポーレート速度を 38400 に変更するには、次のコマンドを使用します。

```
# 38400          :bf:
ttymodes="2502:1805:bf:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

- x86 ベースのシステム: BIOS のシリアル切り替えが有効である場合にコンソール速度を変更します。

システム電源サービスの管理

Oracle Solaris 11 リリースでは、電源管理構成が SMF 構成リポジトリに移動しました。電源関連のコマンド、デーモン、および構成ファイルを組み合わせる代わりに、新しい `poweradm` コマンドを使用してシステムの電源管理プロパティを直接管理します。これらの変更は、Oracle Solaris 11 で電源管理フレームワークを刷新するために広範囲で行われた変更の一部です。

次の電源管理機能は使用できなくなりました。

- `/etc/power.conf`
- `pmconfig` および `powerd`
- デバイス電源管理

次のプロパティは、電源管理コンポーネントを記述します。

- `administrative-authority` – Oracle Solaris 電源管理の管理制御のソースを定義します。このプロパティは `none`、`platform` (デフォルト値)、または `smf` に設定できます。`platform` に設定した場合、`time-to-full-capacity` と `time-to-minimum-responsiveness` の値はプラットフォームの電源管理コマンドから取得されます。`smf` に設定した場合、`time-to-full-capacity` と `time-to-minimum-responsiveness` の値は SMF から取得されます。`time-to-full-capacity` または `time-to-minimum-responsiveness` を逆の設定でプラットフォームのコマンドまたは SMF サービスプロパティから設定しようとすると、値は無視されます。

`administrative-authority` を `none` に設定した場合、Solaris インスタンス内の電源管理はオフになります。

- `time-to-full-capacity` – システムがアクティブなままで低稼働状態または応答性の低い状態からフル稼働状態に達することが許可される最大時間 (ミリ秒単位) を定義します。最大時間には、この範囲内で PM 機能のいずれかまたはすべてを使用しているときに、システムが低稼働状態または応答性の低い状態から戻ってフル稼働状態に達する (再プロビジョニングして使用可能にする) ことが許可される時間が含まれます。

`administrative-authority` のデフォルト設定は `platform` に設定されているため、デフォルトでは、この値はプラットフォーム (たとえば `i86pc`) から取得されます。

または、`administrative-authority` が `smf` に設定されている場合、この値は SMF 電源サービスによって提供される定義から取得されます。インストール時は、この値は未定義です。このプロパティを変更する場合は、システムのワークロードやアプリケーションのニーズに応じて適切な値を検討するようにしてください。

- `time-to-minimum-responsiveness` – システムがアクティブな状態に戻ることが許可される時間 (ミリ秒単位) を定義します。このパラメータは、`time-to-full-capacity` 制約を満たすために必要な最小容量を指定します。`administrative-authority` のデフォルト設定は `platform` に設定されているため、デフォルトでは、このパラメータ値はプラットフォーム (たとえば `i86pc`) から取得されます。

または、`administrative-authority` が `smf` に設定されている場合、この値は SMF 電源サービスによって提供される定義から取得されます。インストール時は、この値は未定義です。このプロパティを変更する場合は、システムのワークロードやアプリケーションのニーズに応じて適切な値を検討するようにしてください。

中程度の値 (たとえば秒単位) を使用すると、プラットフォーム上のハードウェアコンポーネントやサブシステムをより応答性の低い休止状態にすることができません。より大きい値 (たとえば 30 秒から分単位) を使用すると、RAM への保存停止などの手法を使用してシステム全体の保存停止が可能になります。

- `suspend-enable` – デフォルトでは、Solaris を実行しているシステムで保存停止操作の試行は許可されません。このプロパティを `true` に設定すると、保存停止操作の試行が許可されます。`administrative-authority` の値はこのプロパティには影響を与えません。
- `platform-disabled` – `platform-disabled` を `true` に設定すると、プラットフォームの電源管理は無効になります。デフォルト値の `false` に設定すると、電源管理は上記のプロパティの値によって制御されます。

次のコマンドを使用して、電源管理ステータスの簡単なサマリーを表示できます。

```
$ /usr/sbin/poweradm show
Power management is enabled with the hardware platform as the authority:
time-to-full-capacity set to 250 microseconds
time-to-minimum-responsiveness set to 0 milliseconds
```

次のコマンドを使用して、すべての電源管理プロパティを表示できます。

```
$ /usr/sbin/poweradm list
active_config/time-to-full-capacity           current=250, platform=250
active_config/time-to-minimum-responsiveness  current=0, platform=0
active_control/administrative-authority      current=platform, smf=platform
suspend/suspend-enable                       current=false
platform-disabled                             current=false
```

上記の出力で、`active_control/administrative-authority` は2つの設定がある構成のソースを示します。

- platform- プラットフォームに由来する電源管理の構成です。これがデフォルト値です。
- smf - poweradm コマンドを使用してほかの電源管理プロパティを設定できます。

上記の出力の platform-disabled プロパティは、プラットフォームの電源管理が有効になっていることを示しています。

```
platform-disabled          current=false
```

詳細は、[poweradm\(1M\)](#)を参照してください。

例 5-1 電源管理有効化と無効化

以前にシステムの保存停止と復元再開を行うために /etc/power.conf ファイルで S3-support を有効にしていた場合、類似する poweradm の構文は:

```
# poweradm set suspend-enable=true
```

suspend-enable プロパティはデフォルトで false に設定されます。

電源管理を無効にするには、次の構文を使用します。

```
# poweradm set administrative-authority=none
```

次の SMF 電源管理サービスを無効にしても、電源管理は無効になりません。

```
online          Sep_02   svc:/system/power:default
```

保存停止と復元再開を無効にするには、次の構文を使用します。

```
# poweradm set suspend-enable=false
```

例 5-2 電源管理パラメータの設定および表示

次の例は、time-to-full-capacity を 300 ミリ秒、time-to-minimum-responsiveness を 500 ミリ秒に設定する方法を示しています。最後に、新しい値が Oracle Solaris インスタンスに通知されます。

```
# poweradm set time-to-full-capacity=300
# poweradm set time-to-minimum-responsiveness=500
# poweradm set administrative-authority=smf
```

次のコマンドは、現在の time-to-full-capacity の値を表示します。

```
# poweradm get time-to-full-capacity
300
```

次のコマンドは、プラットフォームによって設定された time-to-full-capacity の値を取得します。

```
# poweradm get -a platform time-to-full-capacity
```

例 5-2 電源管理パラメータの設定および表示 (続き)

この値が現在の値と同じになるのは、`administrative-authority` が `platform` に設定されている場合だけです。詳細については、上記の `administrative-authority` プロパティの説明を参照してください。

システム電源の問題のトラブルシューティング

▼ 保守モードの電源サービスから回復する方法

`time-to-full-capacity` と `time-to-minimum-responsiveness` の両方を設定する前に `administrative-authority` を `smf` に設定すると、サービスが保守モードに移行します。このシナリオから回復するには、次の手順を参照してください。

- 1 管理者になります。

『Oracle Solaris 11.1 の管理: セキュリティサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 `administrative-authority` を `none` に設定します。

```
# poweradm set administrative-authority=none
```

- 3 `time-to-full-capacity` と `time-to-minimum-responsiveness` の両方を必要な値に設定します。

```
# poweradm set time-to-full-capacity=value  
# poweradm set time-to-minimum-responsiveness=value
```

- 4 サービスをクリアします。

```
# svcadm clear power
```

- 5 `administrative-authority` を `smf` に設定します。

```
# poweradm set administrative-authority=smf
```

索引

A

- at.deny ファイル, 98, 101
 - 説明, 83
- atjobs ディレクトリ, 88
 - 説明, 83
- at コマンド, 97, 98, 101
 - l オプション (リスト), 100
 - m オプション (メール), 98, 99
 - エラーメッセージ, 102
 - 概要, 83, 84, 97
 - 自動スケジューリング, 88
 - 使用の拒否, 101-102
 - 使用の制御, 98, 101
 - 概要, 83
- at ジョブファイル, 97, 101
 - 書き込み, 97
 - 削除, 101
 - 作成, 98, 99
 - 説明, 85
 - 場所, 85
 - 表示, 100

C

- CPU (中央処理装置)
 - 情報の表示
 - 使用時間, 30, 46
 - 大量に使用するプロセス, 46
- cron.allow ファイル, 93, 95
- cron.deny ファイル, 93, 94
 - デフォルト, 93

- crontab コマンド, 93
 - cron デーモンと, 87
 - e オプション (編集), 89
 - l オプション (リスト), 90, 91
 - r オプション (削除), 92
 - エラーメッセージ, 95
 - 概要, 83, 84
 - 使用されるファイル, 87
 - 使用の制御, 93, 95
 - overview, 93
 - 概要, 83, 93
 - 使用の拒否, 93, 94
 - 使用を特定のユーザーに限定する, 93, 95
 - スケジューリング, 87
 - 変更を保存せずに終了, 89
 - 毎日のタスク, 84
- crontab ファイル
 - 構文, 88
 - 削除, 92
 - 作成, 89, 90
 - 作成と編集, 85-86
 - 使用の拒否, 94
 - 説明, 87, 88
 - デフォルト, 87
 - 場所, 87
 - 表示, 90, 91-92
 - 編集, 89, 90
- cron デーモン, 85, 87

D

df コマンド, 58
-k オプション (キロバイト), 58
概要, 58
例, 58
dispadmin コマンド, 概要, 39

E

eeprom コマンド, ttymon 端末でのボーレートの設定に使用, 106
/etc/cron.d/at.deny ファイル, 98, 101
/etc/cron.d/cron.allow ファイル, 93, 95
/etc/cron.d/cron.deny ファイル, 93, 94

F

fcntl 情報, 32, 34
fsck コマンド, 84
fstat 情報, 32, 34

I

iostat コマンド
概要, 56
基本情報の表示, 56

K

klwp 構造体, 49
kthread 構造体, 49

L

LWP (軽量プロセス)
構造体, 49
情報の表示, 32
定義, 48
プロセスおよび, 48, 49

M

MOTD (その日のメッセージ) 機能, 26
motd ファイル, 26

N

nice コマンド, 44, 45, 46
nice 値, 30, 45

P

perf ファイル, 79
pfiles コマンド, 32, 34
pflags コマンド, 32
pkill コマンド, 32, 35
pldd コマンド, 32
pmap コマンド, 32
pmap コマンド, 32
priocntl コマンド
-c オプション (スケジューリングクラスの指定), 43
-i オプション (ID タイプ), 43
-l オプション (スケジューリングクラスの表示), 40
-m オプション (最高/最低の優先順位), 43
-p オプション (優先順位の指定), 43
-s オプション (優先順位の上限/優先順位の変更), 43
-s オプション (優先順の上限/変更), 43
概要, 39
PROCFS (プロセスファイルシステム), 31
proc 構造体, 30, 49
/proc ディレクトリ, 31
prtconf コマンド, 16
システムの製品名を表示する, 15-16
psig コマンド, 32
psrinfo コマンドオプションによるチップマルチスレッド化機能の識別, psrinfo -p, 22
pstack コマンド, 32
ps コマンド, 30, 33
-c オプション (スケジューリングクラス), 30, 46
-ecl オプション (グローバル優先順位), 40

ps コマンド (続き)

- ef オプション (完全な情報), 32, 33

- 概要, 30

- 報告されるフィールド, 30

ptime コマンド, 32

ptree コマンド, 32, 34

pwait コマンド, 32

pwdx コマンド, 32, 34

S

sa1 コマンド, 78

sa2 コマンド, 78, 79

sadc コマンド, 78, 79

sadd ファイル, 79

sar コマンド, 62, 79

- A オプション (全体のパフォーマンス), 78, 80

- a オプション (ファイルアクセス), 62

- b オプション (バッファ), 63

- c オプション (システムコール), 65

- e オプション (終了時間), 80

- f オプション (データを抽出するファイル), 80

- i オプション (間隔), 80

- m オプション (プロセス間通信), 70

- p オプション (ページイン/ページ障害), 71

- q オプション (待ち行列), 72, 73

- r オプション (未使用のメモリー), 73

- s オプション (開始時間), 80

- u オプション (CPU 使用状況), 74

- v オプション (システムテーブル), 75

- y オプション (端末デバイス), 77

- オプションのリスト, 80

- 概要, 61, 79

- すべてのオプションの説明, 80

svcadm enable system/sar:default コマンド, 79

sys crontab, 79

T

ttymon コンソールの端末でのボーレートの設定,
方法, 106-107

U

/usr/proc/bin ディレクトリ, 31, 32

V

/var/adm/sa/sadd ファイル, 79

/var/spool/cron/atjobs ディレクトリ, 83, 85, 88

/var/spool/cron/crontabs/root ファイル, 86

/var/spool/cron/crontabs/sys crontab, 79

/var/spool/cron/crontabs ディレクトリ, 87

vmstat コマンド, 概要, 52

あ

アドレス空間マップ, 32

アプリケーションスレッド, 48, 50

え

エラーメッセージ

- at コマンド, 102

- crontab コマンド, 95

か

カーネルスレッド

- 構造体, 30, 49

- スケジューリング, 30

き

共有メモリー, プロセスの仮想メモリー, 50

く

繰り返されるシステムタスク, 93

グローバル優先順位

- 定義, 38

- 表示, 40

こ

- コアファイル,自動的に削除する, 97
- コンソールの端末,ボーレートの設定方法, 106-107
- コンソールの端末のボーレート,eeprom コマンドを使用した設定, 106

さ

- 再開,プロセス, 32
- 最小,nice 値, 45
- 最大,nice 値, 45
- 削除
 - at ジョブ, 101
 - crontab ファイル, 92
 - 古い/未使用のファイル, 84
 - ログファイル, 90
- 作成
 - at ジョブ, 98
 - at ジョブ, 99
 - crontab ファイル, 89,90

し

時間

- CPU 時間が大量に増えているプロセス, 46
- CPU 使用, 30
- CPU の使用, 46
- システムタスク
 - 「crontab コマンド、at コマンド」も参照
 - 自動的にスケジュールを設定する, 83
 - スケジューリング
 - 1 回限りのタスク, 84,97
 - 繰り返されるタスク, 84,86
- システムタスクの自動実行, 83
 - 1 回限りのタスク, 97,98,101
 - 繰り返されるタスク, 93,95
- システム動作
 - 追跡されるシステム動作のリスト, 50
 - データの自動収集, 78,79
 - データの手動収集, 79
- システム動作データの自動収集, 78,79
- システム動作レポートの自動生成, 78,79

- システムの製品名,prtconf コマンドを使用して表示する, 15-16
- システムの物理プロセッサタイプ,psrinfo -p, 22-23
- システムリソース
 - 概要, 48
 - 監視, 98
 - 自動, 98
- 新機能,svcadm enable system/sar:default コマンド, 79

す

- スケジューリング
 - 「crontab コマンド、at コマンド」も参照
 - 1 回限りのシステムタスク, 84,97
 - 繰り返されるシステムタスク, 84,86
- スケジューリングクラス, 38
 - 指定, 42
 - 情報の表示, 30,40
 - 変更, 43
 - 優先順位の変更, 43,45
 - 優先順位レベルと, 38,43

せ

制御

- 使用 at コマンド, 83,98,101
- 使用 crontab コマンド, 93,95
 - 概要, 83
 - プロセス, 35
- 制御がきかなくなったプロセス, 46
- 製品名情報を表示する,prtconf コマンド, 15-16
- セキュリティー
 - at コマンド, 98
 - crontab コマンド, 93

そ

- その日のメッセージ (MOTD) 機能, 26

た

- タイムシェアリングプロセス
 - スケジューリングパラメータの変更, 43
- 優先順位
 - 概要, 38
 - 範囲, 38
 - 変更, 43, 44, 45
- 端末, プロセス制御, 30

つ

- ツール
 - システムパフォーマンス監視, 51
 - プロセス, 32
 - プロセス情報の表示, 31

て

- 定型タスクを自動的に実行する (概要), 83
- 停止, プロセスを一時的に, 32
- ディスクドライブ
 - 情報の表示
 - 空きディスク容量, 58
 - 古い/使用されていないファイルの検索と削除, 90
- ディスク容量
 - 情報の表示
 - df コマンド, 58
 - マウントポイント, 59
- ディレクトリ
 - プロセスの現在の作業ディレクトリ, 32
- デフォルト, nice 値, 45

と

- トラブルシューティング
 - プロセス, 46

は

- パフォーマンス
 - 監視ツール, 51
 - システム動作の監視, 50, 62, 78
 - 追跡される動作, 50
 - 動作データの自動収集, 78, 79
 - 動作データの手動収集, 62, 79
 - ファイルアクセス, 62
 - プロセス管理, 32, 45, 48
 - レポート, 61

ひ

- 表示
 - at ジョブ, 100
 - crontab ファイル, 90, 91-92
 - LWP 情報, 32
 - システム情報
 - コマンド, 14
 - システム動作情報, 61, 79
 - システムにインストールされているメモリー, 16
 - 実行中のプロセス, 33
 - スケジューリングクラス情報, 30, 40
 - 日付と時間, 14
 - プロセス, 32
 - プロセス情報 (方法), 33-34
 - ホスト ID, 14
 - 優先順位情報, 30, 40
 - リンクされたライブラリ, 32

ふ

- ファイル
 - fstat および fcntl 情報の表示, 32, 34
 - fstat と fcntl 情報の表示, 32
 - アクセス操作のチェック, 62
- ファイルシステム
 - ディスク容量の使用状況, 58
 - マウントポイント, 59
- ブート, ブート時の sadc コマンドの実行, 79
- フラグの追跡, 32
- プログラム, ディスク依存度, 62

プロセス

- nice 値, 30, 44, 45, 46
- proc ツールコマンド, 31
- proc ツールコマンドを使用した情報の表示, 32
- proc ツールを使用した情報の表示, 31
- trees, 32
- アドレス空間マップ, 32
- アプリケーションスレッドおよび, 48, 50
- 一時的に停止, 32
- 現在の作業ディレクトリ, 32, 34
- 構造体, 30, 49
- 再開, 32
- シグナル動作, 32
- 終了, 32, 35
- 情報の表示, 30
 - LWP, 32
 - priocntl コマンド, 40
 - ps コマンド, 30, 33, 40
 - 実行中のプロセスの表示, 33
 - プロセスの表示, 32
- 情報の表示 (方法), 33-34
- スケジューリングクラス, 38
 - 指定, 42
 - 情報の表示, 30, 40
 - 変更, 43
 - 優先順位の変更, 43, 45
 - 優先順位レベルと, 38, 43
- スタックトレース, 32
- 制御, 35
- 制御がきかない, 46
- ツールコマンド, 32
- ツリー, 32, 34
- 定義, 48
- トラブルシューティング, 46
- 開いているファイルの fstat と fcntl の情報, 32, 34
- フラグの追跡, 32
- 優先順位, 45
 - 概要, 38, 45
 - グローバル優先順位, 38, 40
 - 指定, 43
 - 指定する, 42
 - 情報の表示, 30, 40
 - スケジューリングクラスおよび, 38

プロセス, 優先順位 (続き)

- スケジューリングクラスと, 43
- タイムシェアリングプロセスの優先順位の変更, 43, 44, 45
- 変更, 43, 45
- ユーザーモードの優先順位, 38
- 用語, 48, 50
- リンクされたライブラリ, 32
- プロセスの終了, 32, 35
- プロセスファイルシステム (PROCFS), 31

へ

変更

- crontab ファイル, 89
- スケジューリングクラス, 43
- 日付, 25
- 優先順位, 43, 45
- タイムシェアリングプロセス, 44, 45

編集

- crontab ファイル, 89, 90

ほ

ボーレート

- EEPROM コマンドを使用した設定方法, 106
- TTY 端末での設定方法, 106-107

ま

- 毎週のタスク (crontab を使用したスケジューリング), 84
- 毎月のタスク (crontab を使用したスケジューリング), 84
- 毎日のタスク (crontab を使用したスケジューリング), 84

め

メモリー

仮想

プロセス, 50

共有

プロセスの仮想メモリー, 50

情報の表示例, 16

プロセス構造体および, 49

ゆ

ユーザー構造体, 49

ユーザープロセス

優先順位, 38

優先順位の変更, 44, 45

ユーザーモードの優先順位, 38

優先順位(プロセス)

概要, 38, 45

グローバル

定義, 38

表示, 40

指定, 42, 43

情報の表示, 30, 40

スケジューリングクラスと, 43

変更, 43, 45

タイムシェアリングプロセス, 43, 44, 45

ユーザーモードの優先順位, 38

り

リアルタイムプロセス, クラスの変更, 43

ろ

ログファイル, 自動的に削除, 90

